Programming Guide

This guide describes how to program the HP 53150A, 53151A, and 53152A Microwave Frequency Counters. The information in this guide applies to instruments having the number prefix listed below, unless accompanied by a "Manual Updating Changes" package indicating otherwise.

SERIAL PREFIX NUMBER:

3735A and above (53150A)
3736A and above (53151A)
3737A and above (53152A)

HP 53150A/151A/152A Microwave Frequency Counter

Printed: November 1997

Printed in USA

Manual part number
53150-90002

Certification and Warranty

Certification

Hewlett-Packard Company certifies that this product met its published specification at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

HP warrants HP hardware, accessories and supplies against defects in materials and workmanship for a period of one year from date of shipment. If HP receives notice of such defects during the warranty period, HP will, at its option, either repair or replace products which prove to be defective. Replacement products may be either new or like-new.

HP warrants that HP software will not fail to execute its programming instructions, for the period specified above, due to defects in material and workmanship when properly installed and used. If HP receives notice of such defects during the warranty period, HP will replace software media which does not execute its programming instructions due to such defects.

For detailed warranty information, see back matter.

Safety Considerations

General

This product and related documentation must be reviewed for familiarization with this safety markings and instructions before operation.

Before Cleaning

Disconnect the product from operating power before cleaning.

Warning Symbols That May Be Used In This Book

Instruction manual symbol; the product will be marked with this symbol when it is necessary for the user to refer to the instruction manual.

Indicates hazardous voltages.

Indicates earth (ground) terminal.

or

Indicates terminal is connected to chassis when such connection is not apparent.

Indicates Alternating current.

Indicates Direct current.

Safety Considerations (contd)

WARNING

BODILY INJURY OR DEATH MAY RESULT FROM FAILURE TO HEED A WARNING. DO NOT PROCEED BEYOND A WARNING UNTIL THE INDICATED CONDITIONS ARE FULLY UNDERSTOOD AND MET.

CAUTION

Damage to equipment, or incorrect measurement data, may result from failure to heed a caution. Do not proceed beyond a *CAUTION* until the indicated conditions are fully understood and met.

Safety Earth Ground

An uninterruptible safety earth ground must be maintained from the mains power source to the product's ground circuitry.

WARNING

WHEN MEASURING POWER LINE SIGNALS, BE EXTREMELY CAREFUL AND ALWAYS USE A STEP-DOWN ISOLATION TRANSFORMER WHICH OUTPUT IS COMPATIBLE WITH THE INPUT MEASUREMENT CAPABILITIES OF THIS PRODUCT. THIS PRODUCT'S FRONT AND REAR PANELS ARE TYPCIALLY AT EARTH GROUND. *THUS, NEVER TRY TO MEASURE AC POWER LINE SIGNALS WITHOUT AN ISOLATION TRANSFORMER.*

For additional safety and acoustic noise information, see back matter.

Hewlett-Packard Company
Santa Clara Division
5301 Stevens Creek Boulevard
Santa Clara, California 95052-8059

Contents

1

Before You Start ...

Introduction

This programming guide contains programming information for the HP 53150A, 53151A, and 53152A Microwave Frequency Counters.

This guide assumes you are familiar with the front-panel operation of the Counter. See the *HP 53150A / 151A / 152A Operating Guide* for detailed information about front-panel operation. You should use this programming guide together with tbe operating guide. Knowing how to control the Counter from the front panel and understanding the measurements you want to perform makes the programming task much easier. The operating guide provides explanations and procedures for all of the Counter's measurement functions and contains the specifications for the Counter.

By sending Standard Commands for Programmable Instruments (SCPI) commands, you can remotely operate many of the Counter's front-panel functions via the Hewlett-Packard Interface Bus (HP-IB) or the RS-232 serial interface. These programming commands conform to the *Standard Commands for Programmable Instruments (SCPI) Standard Version 1992.0*. The SCPI standard does not completely redefine how to program instruments over the Hewlett-Packard Interface Bus (HP-IB) or the RS-232 serial interface. However, it does standardize the structure and content of an instrument's command set to reflect the best programming practices developed by people using HP-IB. It also establishes standard command mnemonics for similar functions in all of the instruments that conform to the SCPI standard.

If you have programmed any HP instruments that have been released over the last few years, you have probably seen a general trend toward the techniques specified in the SCPI standard. For example, several instruments are already using a hierarchy of commands that is similar to the command structure defined by the SCPI standard.

Getting Started

Before attempting to program the Counter, take some time to familiarize yourself with the content of this guide. The remainder of this chapter contains the following information:

- An explanation of how you should use the programming guide based on your experience programming instruments and your testing requirements.
- A description of the guide contents.
- A statement of assumptions that are made in the guide.
- A list of related documentation.

How to Use This Guide

How you use this guide depends upon how much you already know about programming instruments and how complex your measurement requirements are. Let's start by establishing your programming background and then discuss the type of measurements you want to perform.

NOTE

With two minor exceptions, the only difference between programming the Counter using the HP-IB interface and the RS-232 serial interface is the manner in which you connect the Counter to the computer. These exceptions are:

1. The Counter sends a command prompt over the RS-232 interface (but not the HP-IB) after receiving and executing each command.

2. When an error is detected (during the Self-Test or during operation), the Counter automatically sends an error message (or messages) over the RS-232 interface (error messages must be requested over the HP-IB). For additional information on error messages, see Appendix B of the *HP 3150A / 151A / 152A Operating Guide*.

New Users

What You Should Understand

As a new user, you must have some understanding of a high-level language, such as BASIC or C, before you can use the command set defined in this guide to control the Counter. (In Chapter 3, "Programming Your Counter for Remote Operation," there are programming examples provided in HP BASIC, Microsoft® QuickBASIC, and Borland® Turbo C.) However, whatever language you use, the command strings that control the Counter remain the same.

Learning to Program the Counter

To learn how to program the Counter, perform the following:

- Scan the summary tables in Chapter 2, "Command Summary," to get a feeling for the number and structure of commands available to you.
- Read and study map drawings in the section titled "Front Panel to SCPI Command Map" in Chapter 2.
- Read Chapter 3, "Programming Your Counter for Remote Operation," for an overview of SCPI concepts as they relate to the HP 53150A, 53151A, and 53152A Frequency Counters. Look at the flowcharts, which illustrate some of the decisions you must make when programming the Counter.
- Read the section at the end of Chapter 3 titled "Programming Examples."
- Modify some of the programming examples to select specific measurement functions. If the programs work, consider yourself an experienced programmer and use Chapter 4, "Command Reference," as a reference for detailed information of all the Counter's SCPI commands.

Experienced Programmers

If you have programmed other HP-IB instruments, you are probably familiar with many of the concepts and tecbniques discussed in this guide. Using the SCPI commands is also very similar to using the earlier HP-IB commands. The main difference between the two command sets is the hierarchy of the subsystem commands. (However, this type of structure has previously been used on other instruments.)

Because the SCPI command set and some of tbe status reporting techniques are new, we advise you to use the following sequence to learn the Counter programming requirements:

- Look over the steps for a new user, and perform any that you think are applicable to your current level of knowledge. In particular, look at the measurement techniques and examples provided in Chapter 3, "Programming Your Counter for Remote Operation."
- Review the summary tables in Chapter 2, "Command Summary." If this chapter contains sufficient information to get you started, write some test programs to explore the Counter's capabilities. If you need additional information on any command, refer to the applicable command description in Chapter 4, "Command Reference ."
- Review the remaining information in this guide to determine what is applicable to your programming requirements.

If you need more information than is contained in this guide, see the section in this chapter titled "Related Documentation."

Applications

After you have read the appropriate information and written some measurement programs, you may want to expand the scope of your applications. The following two techniques are explained in detail:

- If you are going to write interrupt-driven programs (or if you just want to determine the status of the Counter), read the section titled "Status Reporting" in Chapter 3.
- If you are going to write programs to transfer data between the Counter and an external computer, read the section titled "Overview of Response Message Formats" in Chapter 3.

Programming Guide Contents

The following information is contained in this guide:

- Table of Contents
- Chapter 1 (this chapter),"Before You Start ...," is a preface that introduces you to the programming guide.
- Chapter 2, "Command Summary," is a quick reference that summarizes the Counter's programming commands. It provides you with front-panel to SCPI command maps, SCPI conformance information, and command-summary tables.
- Chapter 3, "Programming Your Counter for Remote Operation," describes how to connect and set up the Counter for remote operation, briefly explains the SCPI elements and formats, describes status reporting, describes how to write programs, and provides programming examples for each of the main tasks that you want the Counter to perform.
- Chapter 4, "Command Reference," is a command dictionary that describes the SCPI subsystems and IEEE 488.2 Common commands.
- Chapter 5, "Errors," lists all of the error messages the Counter generates and the cause(s) for each error.
- Index

Assumptions

This guide assumes the Counter is correctly installed and interfaced to an external computer. If it is not, and you intend to use the HP-IB, see the IEEE HP-IB Interconnection information in *Hewlett-Packard Company, Tutorial Description of the Hewlett-Packard Interface Bus, 1987.* (See the section in this chapter titled "Related Documentation" for ordering information.) If you intend to use the RS-232 serial interface, see the section in Chapter 3 titled "To Connect With the RS-232 Serial Interface."

As previously mentioned, this guide also assumes you are familiar with the front-panel operation of the Counter. See the *HP 53150A/151A/152A Operating Guide* for detailed information about front-panel operation. Knowing how to control the Counter from the front panel and understanding the measurements you need to perform makes the programming task much easier.

Related Documentation

This section contains a list of documentation that relates to the use of the Counter. Additional information that may be useful is contained in the following publications:

1. *HP 53150A / 151A / 152A Microwave Frequency Counter Operating Guide* (HP Part Number 53150-90001)
2. *Beginner's Guide to SCPI* (HP Part Number H2325-90002, July 1990 Edition).
3. *Beginner's Guide to SCPI*, Barry Eppler (Hewlett-Packard Press, Addison-Wesley Publishing Co. 1991).
4. *Standard Commands for Programmable Instruments (SCPI)*, (latest version).

   This standard is a guide for the selection of messages to be included in programmable instrumentation. It is primarily intended for instrument firmware engineers. However, you may find it useful if you are programming more than one instrument that claims conformance to the SCPI standard. You can verify the use of standard SCPI commands in different instruments.

   To obtain a copy of this standard, contact:

   SCPI Consortium
   8380 Hercules, Suite P3
   La Mesa, CA 91942
   Phone: (619) 697-8790
   FAX: (619) 697-5955

5. The International Institute of Electrical Engineers and Electronic Engineers, *IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation.*

This standard defines the technical details required to design and build an HP-IB (IEEE 488.1) interface. This standard contains electrical specifications and information on protocol that is beyond the needs of most programmers. However, it can he useful to clarify formal definitions of certain terms used in related documents.

To obtain a copy of this standard, write to:

Institute of Electrical and Electronic Engineers Inc.
345 East 47th Street
New York, NY 10017 USA

6. The International Institute of Electrical Engineers and Electronic Engineers, *IEEE Standard 488.2-1987, IEEE Standard Codes, Formats, Protocols, and Common Commands for Use with ANSI/IEEE Std 488.1-1987 Programmable Instrumentation.*

   This standard defines the underlying message formats and data types used in SCPI. It is intended more for firmware engineers than for instrument users/programmers. However, it can be useful if you need to know the precise definition of specific message formats, data types, or common commands.

   To obtain a copy of this standard, write to:

   The Institute of Electrical and EIectronic Engineers Inc.
   345 East 47th Street
   New York, NY 10017 USA

7. Hewlett-Packard Company, *BASIC 5.0/5.1 Interfacing Techniques Vol 2.*, Specific Interfaces, 1987.

   This HP BASIC manual contains a good non-technicaI description of the HP-IB (IEEE 488.1) interface in Chapter 12, "The HP-IB Interface." Subsequent revisions of HP BASIC may use a slightly different title for this manual or chapter. This manual is the best reference on I/O for HP BASIC programmers.

   To obtain a copy of this manual, contact your nearest Hewlett-Packard Sales office.

8. Hewlett-Packard Company, *Tutorial Description of the Hewlett-Packard Interface Bus*, 1987.

   To obtain a copy of this manual, contact your nearest Hewlett-Packard Sales office.

2

Command Summary

A Quick Reference

Introduction

This chapter is a quick reference that summarizes the Counter's programming commands.

Chapter Summary

1 The section titled "Front Panel to SCPI Command Map" provides maps that show the front-panel keys and their corresponding (or related) SCPI commands.

2 The section titled "HP 53150A/151A/152A Command Summary" lists the IEEE 488.2 Common Commands and SCPI Subsystem commands in Table 2-1 and Table 2-2, respectively.

3 The section titled "*RST Response," lists the states of all of the commands that are affected by the *RST command in Table 2-3. This section also lists commands that are unaffected by *RST in Table 2-4.

Front Panel to SCPI Command Map

Figures 2-1 and 2-2 are command maps that shows the relationships between the front-panel keys and the SCPI commands. This map should help you to identify commands, if you are already familiar with the front panel.

Some SCPI Syntax Conventions:

[ ]	An element inside brackets is optional. Note, the brackets are *not* part of the command and should *not* be sent to the Counter.
1 | 2	Means use either 1 or 2.
<numeric_value>	Means enter a number.
SENSe	Means you *must* use either all the upper case letters or the entire word. The lower case letters are optional. For example, SENS and SENSE are both valid. However, SEN is not valid. (Note SENSe is used here as an example, but this convention applies to all SCPI commands.)

NOTE

When you see quotation marks in a command's parameter (shown in the Parameter Form column in Table 2-2), you must send the quotation marks with the command. Refer to the section titled "Using HP BASIC" on page 3-54 of this guide for details on how to use double quotes or single quotes to enclose the string parameter of a command.

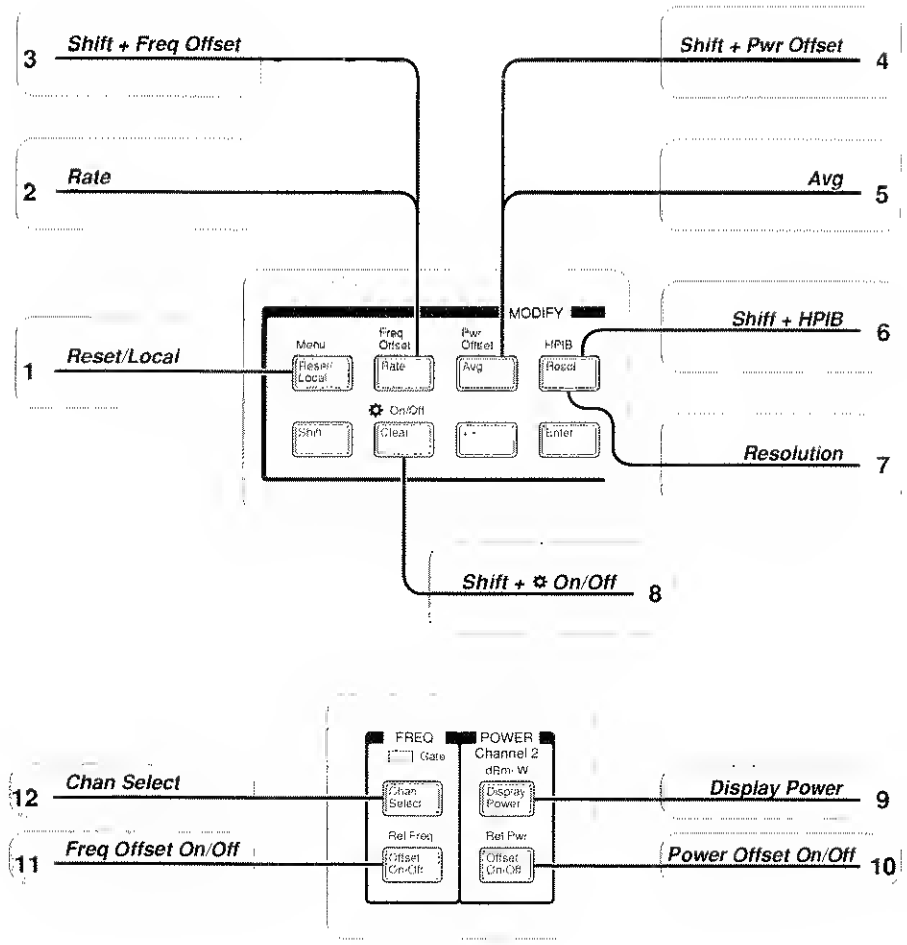


Figure 2-1. Front Panel Control to SCPI Command Map (Part 1 of 2)

1 INITitiate[:IMMediate]

2 TRIGger[:SEQuence]:HOLDoff

3 [SENSe]:FREQuency:OFFSet

4 [SENSe]:POWer:AC:REFerence

5 [SENSe]:AVERage:COUNt

6 [SENSe]:AVERage:STATe

7 SYSTem:COMMunicate:GPIB:ADDRess

8 [SENSe]:FREQuency:RESolution

9 DISPlay:BACKground[:STATe]

10 [SENSe]:FUNCtion

11 [SENSe]:POWer:AC:REFerence:STATe

12 [SENSe]:FREQuency:OFFSet:STATe

13 [SENSe]:FUNCtion

Figure 2-1. Front Panel Control to SCPI Command Map (Part 2 of 2)

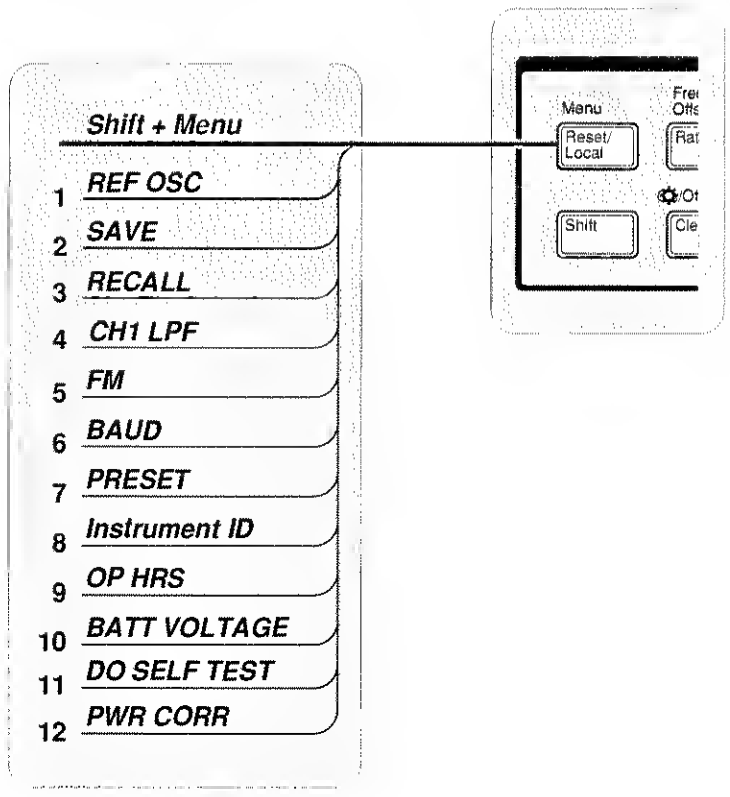


Figure 2-2. Front Panel Menu to SCPI Command Map (Part 1 of 2)

1 [:SENSe]:ROSCillator:SOURce

2 *SAV

3 *RCL

4 :INPut:FILTer[:LPASs][:STATe]

5 [:SENSe]:FILTer:FM:AUTO

6 :SYSTem:COMMunicate:SERial[:RECeive]:BAUD

7 *RST

8 *IDN?

9 See Service Manual

10 See Service Manual

11 *TST?

12 MEMory:CLEar[:NAME]
MEMory:DATA
[:SENSe]:CORRection:CSET:SELect
[:SENSe]:CORRection:CSET:STATe

Figure 2-2. Front Panel Menu to SCPI Command Map (Part 2 of 2)

HP 53150A/151A/152A Command Summary

This section summarizes both the IEEE 488.2 Common and HP 53150A/151A/152A Standard Commands for Programmable Instruments (SCPI) commands in tabular format. IEEE 488.2 Common Commands are listed first, followed by SCPI commands.

SCPI Conformance Information

The SCPI commands used in the HP 53150A/151A/152A Counters are in conformance with the SCPI Standard Version 1995.0. The SCPI command set consists of the following:

- Common Commands as defined in IEEE 488.2-1987—listed and summarized in Tahle 2-1.
- SCPI Suhsystem commands as confirmed (and listed) in the SCPI Standard—the commands defined in Table 2-2 as "Std."
- SCPI Suhsystem commands designed for the instrument in conformance with SCPI standards hut not yet listed in the SCPI Standard—the commands defined in Table 2-2 as "New."
- Details of all HP 53150A/151A/152A commands can be found in Chapter 4, "Command Reference."

Information on the SCPI commands format, syntax, parameter, and response types is provided in Chapter 3, "Programming Your Counter for Remote Operation."

IEEE 488.2 Common Commands

The Common Commands are general-purpose commands that are common to all instruments (as defined in IEEE 488.2). Common Commands are easy to recognize because they all begin with an "*" (for example, *RST, *IDN?, *OPC). These commands are generally not related to measurement configuration. They are used for functions like resetting the instrument, identification, or synchronization.

Table 2-1 lists the IEEE 488.2 Common Commands supported by the HP 53150A/151A/152A in alpbabetical order by mnemonic, name, and function. More information concerning the operation of IEEE 488.2 status-reporting commands and structure can be found in the "Status Reporting" section of Chapter 3. Standard explanations of the IEEE 488.2 Common Commands can be found in the *ANSI/IEEE Std. 488.2-1987, IEEE Standard Codes, Formats, Protocols, and Common Commands* document.

Table 2-1. IEEE 488.2 Common Commands

Mnemonic	Command Name	Function
*CLS	Clear Status	Clears all event status registers summarized in the status byte and empties the Error Queue.
*DDT <arbitrary block>	Define Device Trigger Command	Defines which command is executed when the Counter receives a GET or *TRG command.
*DDT?	Define Device Trigger Query	Queries which command is executed when the Counter receives a GET or *TRG command.
*ESE <NRf>	Standard Event Status Enable	Sets the Standard Event Status Enable Register.
*ESE?	Standard Event Status Enable Query	Queries the Standard Event Status Enable Register.
*ESR?	Event Status Register Query	Queries and then clears the Standard Event Status Register.
*IDN?	Identification Query	Queries the Counter identification.
*IST?	Instrument Status Query	Queries the current state of the parallel poll response (Instrument Status).
*OPC	Operation Complete	Causes Counter to set the operation complete bit in the Standard Event Status Register when all pending operations (see Note at the end of table) are finished.
*OPC?	Operation Complete Query	Places an ASCII "1" in the Output Queue when all pending operations (see Note at the end of table) are completed.
*PRE <NRf>	Parallel Poll Enable Register	Sets the value of the Parallel Poll Enable register.
*PRE?	Parallel Poll Enable Register Query	Queries the value of the Parallel Poll Enable register.
*RCL <NRf>	Recall	Restores the state of the Counter's user settings from a copy stored in local non-volatile memory (0 through 9 are valid memory registers).
*RST	Reset	Resets the Counter to a known state, as defined in this manual.
*SAV <NRf>	Save	Stores the current state of the Counter's user settings in local non-volatile memory (0 through 9 are valid memory registers).
*SRE <NRf>	Service Request Enable	Sets the Service Request Enable register.

Table 2-1. IEEE 488.2 Common Commands (Continued)

Mnemonic	Command Name	Function
*SRE?	Service Request Enable Query	Queries the Service Request Enable register.
*STB?	Status Byte Query	Queries the Status Byte and Master Summary Status bit.
*TRG	Trigger	This trigger command is the device-specific analog of the IEEE 488.1 defined GET. It initiates the action specified by the *DDT command.
*TST?	Self-Test Query	Executes an internal self-test and reports the results.
*WAI	Wait-to-Continue	Makes the Counter wait until all pending operations (see Note) are completed before executing commands that follow the *WAI command.
Note: Pending operations include measurements in progress.		

HP 53150A/151A/152A SCPI Subsystem Commands

SCPI Suhsystem commands include all measurement functions and some general-purpose functions. SCPI Subsystem Commands use a hierarchy relationship hetween keywords that is indicated by a colon (:). For example, in the SYST:ERR? query, the ":" between SYST and ERR? indicates ERR? is subordinate to SYST.

Table 2-2 lists the SCPI Subsystem Commands in alphabetical order by the command keyword. The table shows the Subsystem commands hierarchical relationship, related parameters (if any), and any associated information and comments.

CAUTION

Not all commands have a query form. Unless otherwise stated in Table 2-2, commands have both a command and a query form. Any command in the table that is shown with a "?" at the end, is a "Query Only" command.

Std/New Column

The **Std/New** column in Table 2-2 shows the status of the command with respect to the SCPI standard. The "Std" commands operate as defined in the SCPI standard and as defined in this guide.

The category of "New" consists of commands that could be:

- SCPI approved but are not yet in the SCPI manual
- HP approved and submitted for SCPI approval.
- Not approved at all.

The "New" commands operate as defined in this guide.

Parameter Form Column

Refer to the section titled "Parameter Types" on page 3-16, "Programming Your Counter for Remote Operation," for descriptions of the different parameter types (such as <Boolean>, <NRf>, <arbitrary block>, etc.).

Table 2-2. HP 53150A/151A/152A SCPI Command Summary

Keyword/Syntax	Parameter Form	Std/New	Comments
:ABORt		Std	Event; no query. Resets the trigger system and aborts any measurement in progress. Places the trigger system in the IDLE state.
:CONFigure		Std	See Measurement Instructions in this table.
:DISPlay		Std	Subsystem. Controls the selection and presentation of textual information on the display.
:ENABle	<Boolean>	Std	Controls whether or not the entire display is visible.
[:WINDow]			
:BACKground			
[:STATe]	<Boolean>	New	Turns the LCD display backlight ON or OFF.
:FETCh		Std	See Measurement Instructions in this table.
:INITiate		Std	Subsystem. Controls the initiation of measurements.
:CONTinuous	<Boolean>	Std	Sets the instrument for continuously initiated or user-initiated measurements.
[:IMMediate]		Std	Event; no query. Causes the instrument to initiate and complete one full measurement cycle.
:INPut		Std	Subsystem. Controls the characteristics of the instrument's Channel 1 input port.
:FILTer		Std	Subtree. Controls a filter that can be inserted in the path of the measurement signal.
[:LPASs]		Std	Subtree. Selects the Low-PASs filter.
[:STATe]	<Boolean>	Std	Enables or disables the Channel 1 low-pass filter (approx. 50 KHz).
:MEASure		Std	See Measurement Instructions in this table.

Table 2-2. HP 53150A/151A/152A SCPI Command Summary (Continued)

Keyword/Syntax	Parameter Form	Std/New	Comments
Measurement Instructions			
:CONFigure[:SCALar]:<function>	See <parameters> and <source_list> below.	Std	Configures the instrument to perform the specified measurement.
:CONFigure?		Std	Returns the function configured by the last :CONFigure or :MEASure command.
:MEASure:[:SCALar]:<function>?	See <parameters> and <source_list> below.	Std	Configures the instrument, initiates measurement, and queries for the result (i.e., provides a complete measurement sequence.
:READ[:SCALar]:<function>?		Std	Initiates measurement, and queries for the result. (Performs a :FETCh? on "fresh" data.)
:FETCh[:SCALar]:<function>?		Std	Queries the measurement made by a previous :MEASure, :READ, or :INITiate command.

*The <function> and corresponding <parameters> and <source_list> are defined below:

<function>	**<parameters>**	**[,<source_list>]***	**Std/New**
[:VOLTage]:FREQuency	[<expected_value>[,<resolution>]]	[,(@1) | (@2)]	Std
:POWer[:AC]	[<expected_value>[,<resolution>]]	[,(@2)]	Std

*<source_list> has the same syntax as SCPI <channel_list>. For example, a frequency measurement on channel 2 uses (@2) to specify channel 2.

Table 2-2. HP 53150A/151A/152A SCPI Command Summary (Continued)

Keyword/Syntax	Parameter Form	Std/New	Comments
:MEMory		Std	Subsystem. Manages instrument memory.
:CLEAr[:NAME]	<name>	Std	Event; no query. Restores the frequency values in the named correction profile to the default values and sets all loss values to zero.
:DATA	<name>, <data>	Std	Stores data in the named correction profile.
:DATA?	<name>	Std	Queries the data in the named correction profile.
:NSTates?		Std	Query only. Returns the number of available *SAV/*RCL states in the instrument.
:READ		Std	See Measurement Instructions in this table.
[:SENSe]		Std	Subsystem setup commands.
:AVERage		New	Subtree. Configures the averaging function.
[:STATe]	<Boolean>	New	Turns averaging ON and OFF.
:COUNt	<numeric_value>	New	Specifies the number of measurements to combine when AVERage:STATe is ON.
:CORRection		Std	Subtree. Configures the power-correction function.
:CSET			
:SELect	<character_data> CORR1|CORR2...CORR9	Std	Selects a power-correction profile.
:STATe	<Boolean>	Std	When STATe is ON, power measurements are modified according to the data in the correction profile selected with :SELect.
:DATA?	<data_handle>	Std	Query only. Returns the current measurement result data of the SENSe subsystem.
	"[XNONe]FREQuency [1|2]"	Std	Frequency on channel 1 or 2.
	"[XNONe]POWer [2]"	Std	Power on channel 2.
:FILTer		Std	Subtree. Controls the use of filtering routines in the instrument.
:FM			
:AUTO	<Boolean>	New	Turns automatic FM compensation ON or OFF.

Table 2-2. HP 53150A/151A/152A SCPI Command Summary (Continued)

Keyword/Syntax	Parameter Form	Std/New	Comments
[:SENSe] (cont.)			
:FREQuency		Std	Subtree. Controls the frequency-measuring capabilities of the instrument.
:OFFSet	<numeric_value>[frequency unit]	Std	Sets a reference frequency for all other absolute frequency settings in the instrument.
:STATe	<Boolean>	New	When STATe is ON, frequency measurements are modified by the value of FREQ:OFFset.
:RESolution	<numeric_value>[frequency unit]	Std	Sets the frequency-measurement resolution.
:TRACking	<character_data> FAST|SLOW|OFF	New	Selects one of three signal-tracking modes.
:FUNCtion		Std	Subtree. Selects the <sensor_function>(s) to be sensed by the instrument.
[:OFF]	<sensor_function>[,<sensor_function>] "[XNONe]FREQuency [1|2]" "[XNONe]POWer [2]"	New	Selects the <sensor_function>(s) to be turned OFF.
[:ON]	<sensor_function>[,<sensor_function>] "[XNONe]FREQuency [1|2]" "[XNONe]POWer [2]"	Std	Selects the <sensor_function> to be sensed by the instrument.
:STATe?	<sensor_function>	Std	Query that returns a Boolean value which indicates whether the specified <sensor_function> is ON or OFF.
:POWer		Std	Subtree. Configures the instrument for power measurement on channel 2.
:AC			
:REFerence	<numeric_value>	Std	Sets a reference amplitude (in dB) for display of power measurements.
:STATe	<Boolean>	Std	Determines whether amplitude is measured in absolute or relative mode.
:ROSCillator		Std	Subtree. Controls the reference oscillator.
:SOURce	<character_program_data> INTernal|EXTernal	Std	Sets the selection of a reference timebase (INTernal or EXTernal).

Table 2-2. HP 53150A/151A/152A SCPI Command Summary (Continued)

Keyword/Syntax	Parameter Form	Std/New	Comments
:STATus		Std	Subsystem. Controls the SCPI-defined (Qperation and Questionable) status-reporting structures.
:OPERation		Std	Subtree.
:CQNDition?		Std	Query only. Queries the Qperation Condition Status Register.
:ENABle	<non-decimalnumeric>|<NRf>	Std	Sets the Qperation Event Status Enable Register.
[:EVENt]?		Std	Query only. Queries and then clears the Operation Event Status Register.
:NTRansition	<non-decimalnumeric>|<NRf>	Std	Sets and queries the negative transition filter for the Qperation status reporting structure.
:PTRansition	<non-decimal numeric> | <NRf>	Std	Sets and queries the positive transition filter for the Qperation status reporting structure
:PRESet		Std	Event; No query. Presets the enable registers and transition filters associated with the Operation and Questionable status reporting structures.
:QUEStionable		Std	Subtree.
[:EVENt]?		Std	Query only. Queries and then clears the Questionable Data Event Status Register.
:CONDition?	<non-decimal numeric> | <NRf>	Std	Query only. Queries the Questionable Data Condition Status Register.
:ENABle		Std	Sets the Questionable Data Event Status Enable Register structures.

Table 2-2. HP 53150A/151A/152A SCPI Command Summary (Continued)

Keyword/Syntax	Parameter Form	Std/New	Comments
:SYSTem		Std	Subsystem. Collects the functions that are not related to instrument performance.
:COMMunicate		Std	Subtree. Collects together configuration of control/communication interfaces.
:GPIB		Std	Subtree. Controls the HPIB.
[:SELF]			
:ADDRess	<numeric_value>	Std	Sets the HPIB address of the instrument.
:SERial		Std	Subtree.
[:RECeive]			
:BAUD	<numeric_value>	Std	Sets the baud rate.
:ERRor?		Std	Query only. Queries the oldest error in the Error Queue and removes the error from the queue (first in-first out). See Chapter 5 for error definitions.
:KEY		Std	Simulates the pressing of a front-panel key.
:CATalog?		Std	Query only. Returns a list of defined key codes.
:VERSion?		Std	Query only. Returns the SCPI version number with which the Counter complies.
:TRIGger		Std	Subsystem.
[:SEQuence]			
:HOLDoff	<numeric_value>	Std	When INIT:CONT ON, this command specifies the length of the delay between measurements.

*RST Response

The IEEE 488.2 *RST command returns the instrument to a specified state optimized for remote operation. (Use *CLS to clear the status event registers and the SCPI error queue.)

The states of command settings affected by the *RST command are described in Table 2-3. Table 2-4 lists command settings that are unaffected by *RST.

Table 2-3. HP 53150A/151A/152A *RST State

Command Header	Parameter	State
*DDT	<arbitrary block>	#14INIT
:DISPlay[:WINDow]:BACKground[:STATe] :DISPlay:ENABle	<Boolean> <Boolean>	ON ON
:INITiate:CONTinuous	<Boolean>	OFF
:INPut:FILTer:[:LPASs][:STATe]	<Boolean>	OFF
[:SENSe]:AVERage:[STATe] [:SENSe]:AVERage:[COUNt] [:SENSe]:FILTer:FM:AUTO	<Boolean> <numeric_value> <Boolean>	OFF 1 ON
[:SENSe]:CORRection:CSET:SELect [:SENSe]:CORRection:CSET:STATe [:SENSe]:FREQuency:OFFset [:SENSe]:FREQuency:OFFset:STATe [:SENSe]:FREQuency:RESolution [:SENSe]:FREQuency:TRACking [:SENSe]:FUNCtion:OFF [:SENSe]:FUNCtion[:ON] [:SENSe]:POWer:AC:REFerence [:SENSe]:PQWer:AC:REFerence:STATe [:SENSe]:ROSCillator:SOURce	<character_data> <Boolean> <numeric_value>[frequency unit] <Boolean> <numeric_value>[frequency unit] <character_program_data> <sensor_function> <sensor_function> <numeric_value> <Boolean> INTernal | EXTernal <Boolean>	CORR1 OFF 0 OFF 1 Hz SLOW "FREQuency 1", "POWer 2" "FREQuency 2" 0 OFF INTernal
:TRIGger[:SEQuence]:HOLDoff	<numeric_value>	0

Table 2-4. Unaffected by *RST

Item
*ESE
*PRE
*SRE
:MEMory:NSTates?
:STATus subsystem—all command settings
:SYSTem subsystem—all command settings

3

Programming Your Counter for Remote Operation

Introduction

This chapter provides remote-operation setup and programming information. You can use this information to configure the Counter to operate as a remote device.

NOTE

Most of this chapter deals with programming the HP 53150A/151A/152A Counters using SCPI and IEEE 488.2 commands. With two minor exceptions, the only difference between programming these Counters using the HP-IB interface and the RS-232 serial interface is the manner in which you connect the Counter to the computer. These exceptions are:

1. The Counter sends a command prompt over the RS-232 interface (but not the HP-IB) after receiving and executing each command.

2. When an error is detected (during the Self-Test or during operation), the Counter automatically sends an error message (or messages) over the RS-232 interface (error messages must be requested over the HP-IB). For additional information on error messages, see Appendix B of the *HP 53150A / 151A / 152A Operating Guide*.

Chapter Summary

Where to Find Some Specific Information

Programming Examples

Connecting the Counter to a Computer

To program the Counter to operate remotely, you need to interface the Counter with a computer. The HP 53150A, 53151A, and 53152A Microwave Frequency Counters provide two interfaces for remote, computer-controlled operation—HP-IB and RS-232. The following sections describe how to connect and configure both interfaces for remote Counter operation.

To Connect With the HP-IB

To connect the Counter to a computer using the HP-IB, install an HP-IB cable (such as the HP 10833A cable) between the two units, as shown in Figure 3-1.

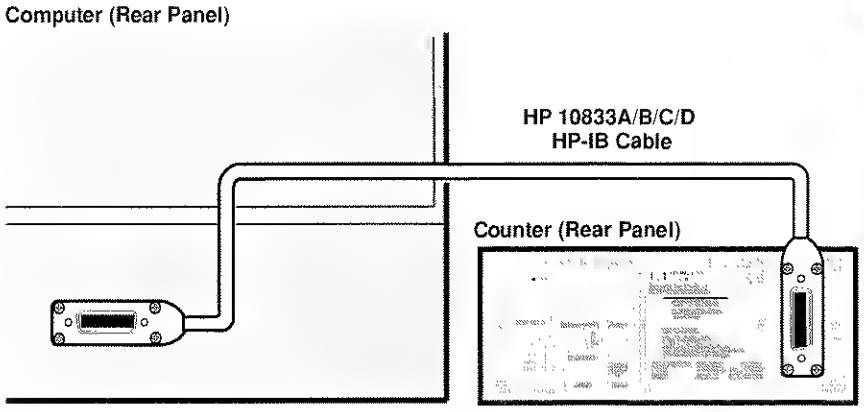


Figure 3-1. HP-IB Interconnection

Configuring the HP-IB

The Counter's HP-IB operates in Addressed (talk/listen) mode, which provides bi-directional communication. The Counter can receive commands and setups from a computer, and it can send data and measurement results. There is one configurable setting related to HP-IB communication—the HP-IB Address.

The following section, titled "Changing the HP-IB Address," provides instructions for setting the HP-IB address from the Counter's front panel.

NOTE

Once the Counter is in Remote mode, all front-panel keys except the **Reset/Local** key are disabled. As long as local-lockout is off, pressing the **Reset/Local** key returns the counter to Local mode.

Changing the HP-IB Address

1 Press and release the **Shift** key, and then press **HPIB (Resol)**. The *HPIB ADDR* menu is displayed, the current HP-IB address is shown to the right of the blinking indicator (>), and the LED indicator between the arrow keys flashes.

2 Press the right-arrow key. The blinking indicator changes direction (from > to <), and the current HP-IB address blinks.

3 Press (or press and hold) tbe up-arrow or down-arrow key to change tbe HP-IB address (the available addresses are 1 to 30).

4 When your desired address is displayed, press the **Enter** key. The address you selected is assigned, and the display returns to its normal operating mode. You *must* press the **Enter** key to complete the entry.

NOTE

To configure the Counter so that the a specific HP-IB address is automatically assigned each time you turn the Counter on, select the address, and then save your current settings in SAV 0. The settings in SAV 0 are recalled each time the Counter is turned on.

IEEE 488.1 Interface Capabilities

The HP 53150A/151A/152A Counter has the following IEEE 488.1 Interface capabilities:

SH1	SR1	DT1
AH1	RL1	C0
T6	PP1	E1
L3	DC1	

To Connect With the RS-232 Serial Interface

The HP 53150A, 53151A, and 53152A Frequency Counters use an RJ12 modular connector for the RS-232 interface. This connector is accessible through the back panel of the counter, as shown in Figure 3-2.

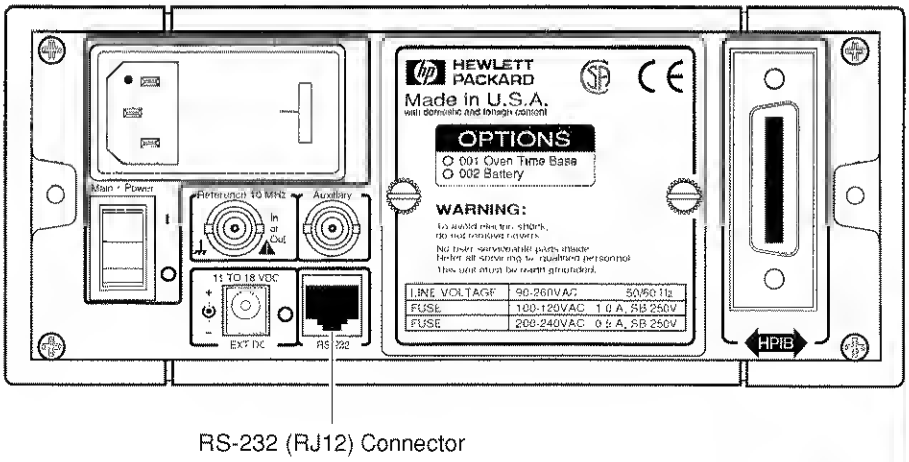


Figure 3-2. Location of the RS-232 (RJ12) Connector

To connect the Counter to a computer using the RS-232 interface, you need a serial cable that has an RJ12 modular connector at the Counter end and a female DB25 connector at the computer end.

Making an RS-232 Cable

Most computers use male DB25 connectors for their serial ports. Therefore, you must use either a cable with an RJ12 plug at the Counter end and a female DB25 connector at the computer end or a double-ended RJ12 cable and an RJ12-to-DB25F adapter to interface the Counter with a computer. Since pre-manufactured RJ12/DB25 cables are rare, it is probably most efficient to obtain the necessary parts, and assemble the cable yourself.

Assembling the DB-25/RJ12 Adapter and the Cable

Use the following procedure to wire the adapter and assemble the cable:

1 Obtain a male DB25 to female RJ12 adapter, such as the Voltrex MAK206F (manufactured by SPC Technology) or equivalent, and either a 6-conductor male-to-male RJ12 cable of a suitable length or a similar length of 6-conductor, flat telephone cable and two RJ12 plugs. RJ12 modular plugs (SPC part number TA30-6) and 6-conductor, flat telephone cable (SPC part number TXW6151) are also available from SPC Technology (and other manufacturers).

2 Adapter kits like the Voltrex MAK206F usually include a pre-wired RJ12 modular receptacle, a DB25F connector, and the adapter body, or wiring shroud. Wire the RJ12 receptacle to the DB25F connector according to the diagrams in Figure 3-3, and then assemble the adapter according to the instructions included in the kit.

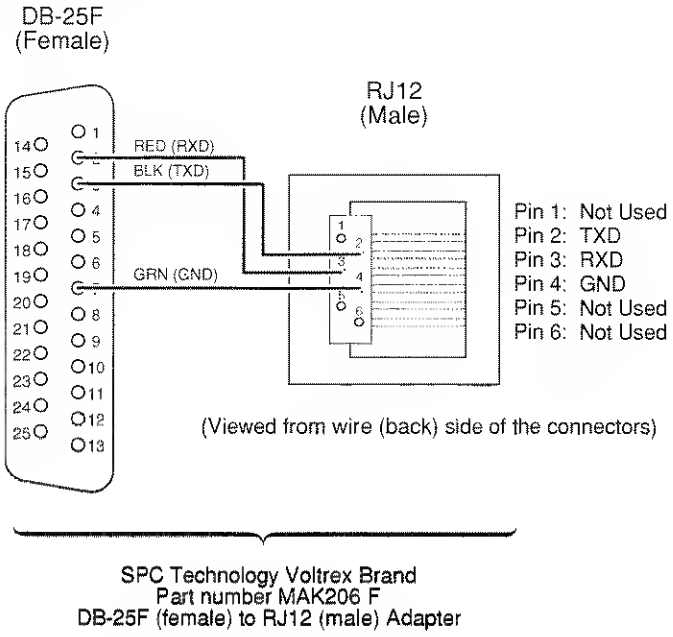


Figure 3-3. .Wiring the RJ12/DB25 Adapter (1 of 2)

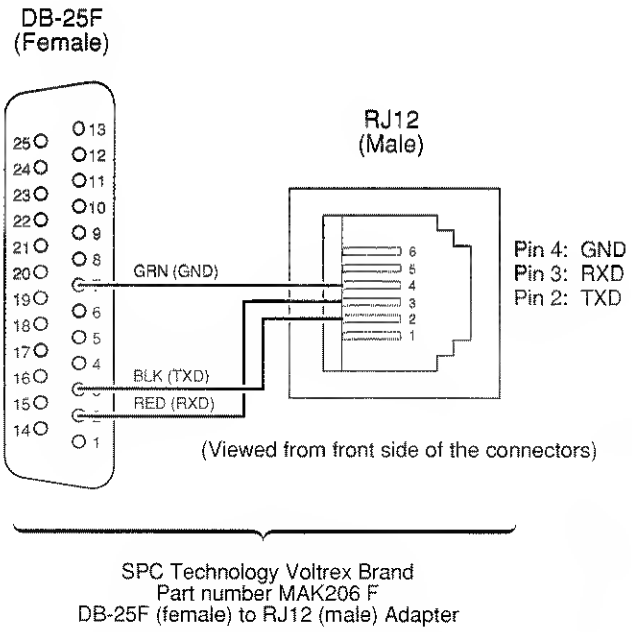


Figure 3-3. Wiring the RJ12/DB25 Adapter (2 of 2)

3 Attach an RJ12 modular plug to each end of a suitable length of 6-conductor, flat telephone cable as shown in Figure 3-4. Be sure to attach the connectors in the orientations shown in the figure.

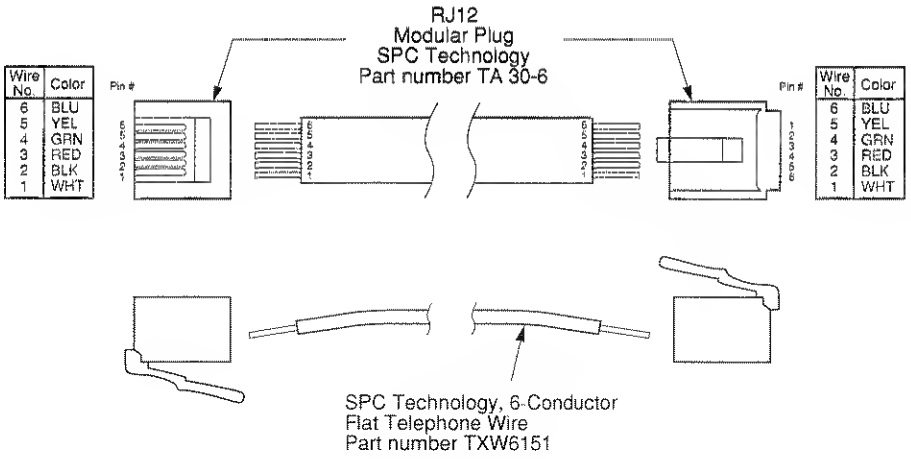


Figure 3-4. Assembling the Cable

4 Connect either end of the cable to the adapter by inserting the RJ12 plug into the receptacle on the adapter.

Connecting with the Serial Interface

Connect the female DB25 connector on the adapter to the male DB25 serial-port connector on the computer, and then insert the RJ12 plug at the other end of the cable into the RJ12 receptacle on the back of the counter as shown in Figure 3-5.

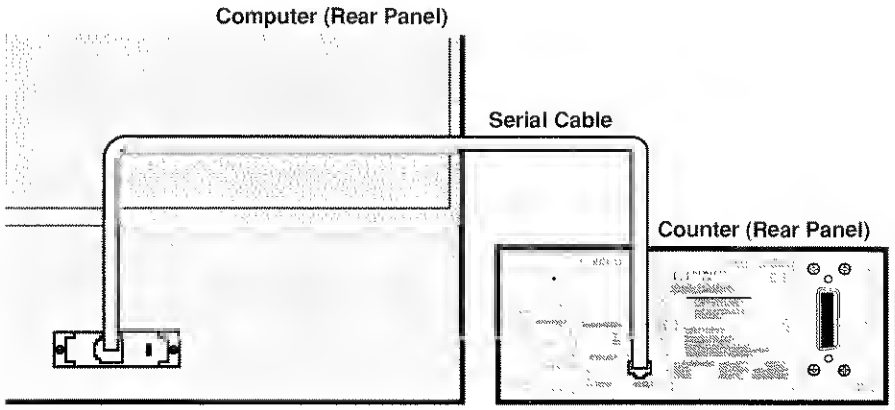


Figure 3-5. RS-232 Serial Interconnection

Remote/Local Operation

When the counter is connected to a computer via the HP-IB, and it is in Remote mode, the **Rmt** indicator is visible on the display, and the Counter settings cannot be affected using the front-panel controls. The **Reset/Local** key can be used to manually return the counter to local control (if local-lockout is off).

When the Counter is in Local mode, the front-panel **Rmt** indicator in the display is off.

Overview of Command Types and Formats

There are two types of HP 53150A/151A/152A programming commands: IEEE 488.2 Common Commands and Standard Commands for Programmable Instruments (SCPI). The format of each type of command is described in the following paragraphs. (Refer to Chapter 2, "Command Summary," for SCPI conformance information.)

Common Command Format

The IEEE 488.2 Standard defines Common Commands as commands that perform functions like reset, self-test, status byte query, and identification. Common Commands always begin with the asterisk (*) character, and may include parameters. The command keyword is separated from the first parameter by a space character. Some examples of Common Commands are as follows:

```
*RST        *IDN?        *RCL 1
```

SCPI Command and Query Format

SCPI commands perform functions like instrument setup. A subsystem command has a hierarchical structure that usually consists of a top level (or root) keyword, one or more lower-level keywords, and parameters. The following example shows a command and its associated query:

```
:DISPlay:ENABle:ON
:DISPlay:ENABle?
```

In this example, DISPlay is the root-level keyword, ENABle is the second-level keyword, and ON is the command parameter.

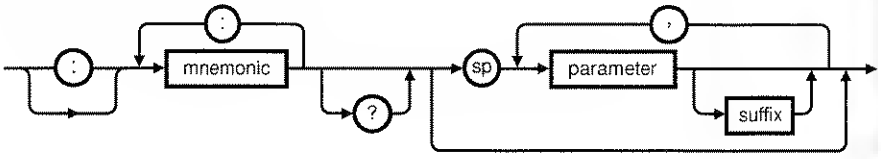
Elements of SCPI Commands

A program command or query is composed of functional elements that include a header (or keywords with colon separators), program data, and terminators. These elements are sent to the Counter over the HP-IB or the RS-232 interface as a sequence of ASCII data messages. Examples of a typical Common Command and Subsystem Command are:

```
OUTPUT 712;"*CLS"
OUTPUT 712;":DISP:ENAB ON;:FREQ:RES 1KHz"
```

Subsystem Command Syntax

Figure 3-6 shows the simplified syntax of a Subsystem Command. You must use a space (SP) between the last command mnemonic and the first parameter in a Subsystem Command. Note that if you send more than one parameter with a single command, you must separate adjacent parameters with a comma.



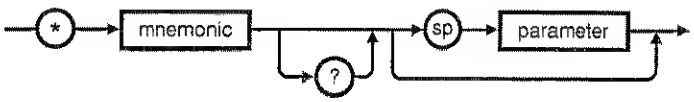
NOTE

NOTE: sp = space. ASCII character decimal 32

Figure 3-6. Simplified Program Command Syntax Diagram

Common Command Syntax

Figure 3-7 shows the simplified syntax of a Common Command. You must use a space (SP) between the command mnemonic and the parameter in a Common Command.



NOTE

NOTE: sp = space. ASCII character decimal 32

Figure 3-7. Simplified Common Command Syntax Diagram

Abbreviated Commands

The command syntax shows most keywords as a mixture of upper- and lowercase letters. Uppercase letters indicate the abbreviated spelling for the command. For better program readability, you may send the entire keyword. The HP 53150A/151A/152A accepts either command form and is not case sensitive.

For example, if the command syntax shows DISPlay, then DISP and DISPLAY are both acceptable forms. Other forms of DISPlay, such as DISPL or DISPLA are illegal, and they generate errors. You may use upper and/or lower case letters. Therefore, DISPLAY, display, and DiSpLaY are all acceptable.

Keyword Separator

A colon (:) always separates one keyword from the next lower-level keyword as shown below:

:DISPlay:ENABle?

Optional Keyword

Optional keywords are those which appear in square brackets ([ ]) in the command syntax. (Note that the brackets are not part of the command and are not sent to the Counter.)

Suppose you send a second level keyword without the preceding optional keyword. In this case, the Counter assumes you intend to use the optional keyword and responds as if you had sent it.

Examine the portion of the [:SENSe] subsystem shown below:

```
[:SENSe]
  :FREQuency
    :RESolution
```

The root-level keyword [:SENSe] is an optional keyword. To set the Counter's frequency resolution, you can use either of the following:

```
:SENS:FREQ:RES
      or
:FREQ:RES
```

Parameter Types

Table 3-1 contains explanations and examples of parameter types. Parameter types may be numeric value, Boolean, literal, NRf, string, non-decimal numeric, or arbitrary block.

Table 3-1. Command and Query Parameter Types

TYPE	EXPLANATIONS AND EXAMPLES
<numeric value>	Accepts all commonly used decimal representation of numbers including optional signs, decimal points, and scientific notation: 123, 123e2, -123, –1.23e2, .123, 1.23e–2, 1.23000E–01. Special cases include MINimum and MAXimum as follows: MINimum selects minimum value available, and MAXimum selects maximum value available. Queries using MINimum or MAXimum return the associated numeric value.
<Boolean>	Represents a single binary condition that is either true or false: 1 or ON, 0 or OFF (Query response returns only 1 or 0.) An <NRf> is rounded to an integer. A non-zero value is interpreted as 1.
<literal>	Selects from a finite number of choices. These parameters use mnemonics to represent each valid setting. An example is the INPut:COUPling AC | DC command parameters (AC | DC).
<NRf>	Flexible numeric representation.
<string>	A string parameter is delimited by either single quotes or double quotes. Within the quotes, any characters in the ASCII 7-bit code may be specified. The following HP BASIC statement sends a command containing a <string> parameter: **OUTPUT 703;"FUNC 'FREQ'"**
<non-decimal numeric>	Format for specifying hexadecimal (#H1F), octal (#Q1077), and binary (#B10101011) numbers using ASCII characters. May be used in :STATus subsystem commands.
<arbitrary block>	The syntax is a pound sign (#) followed by a non-zero digit representing the number of digits in the subsequent decimal integer. The decimal integer specifies the number of 8-bit data bytes being sent. This is followed by the actual data. The terminator is a line feed asserted with EOI. For example, for transmitting 8 bytes of data, the format could be: **Number of digits that follow** ↓ **Actual data** **Terminator** #208<8 bytes of data><new line> ^EOI **Number of bytes to be transmitted** The "2" indicates the number of digits that follow and the two digits "08" indicate the number of *data* bytes to be transmitted; a zero-length block has the format: #0<new line>^EOI; <new line> is defined as a single ASCII-encoded byte corresponding to 10 decimal.

Parameter Separator

If you send more than one parameter with a single command, you must separate adjacent parameters with a comma.

Query Parameters

All selectable <numeric value> parameters (except Common Commands) can be queried to return the minimum, maximum, and DEFault values they are capable of being set to by sending a MINimum, MAXimum, or DEFault parameter after the "?." For example, consider the AVERage:COUNt? query.

If you send the query without specifying a parameter (AVER:COUN?), the present setting is returned. If you send the MIN parameter (using AVER:COUN? MIN), the command returns the minimum acceptable count. If you send the MAX parameter, the command returns the maximum level currently available. Be sure to place a space between the question mark and the parameter.

Suffixes

A suffix is the combination of suffix elements and multipliers that can be used to interpret some <numeric value>. If a suffix is not specified, the Counter assumes that <numeric value> is unscaled (that is, Volts, seconds, etc.)

For example, the following two commands are equivalent:

```
OUTPUT 703;"FREQ:RES 1KHz"
OUTPUT 703;"FREQ:RES 1E+3"
```

Suffix Elements

Suffix elements, such as HZ (Hertz), S (seconds), V (volts), OHM (Ohms), PCT (percent), and DEG (degrees) are allowed within this format.

Suffix Multipliers

Table 3-2 lists the suffix multipliers that can be used with suffix elements (except PCT and DEG).

Table 3-2. Suffix Multipliers

DEFINITION	MNEMONIC	NAME
1E15	PE	PETA
1E12	T	TERA
1E9	G	GIGA
1E6	MA (or M for OHM and HZ)*	MEGA
1E3	K	KILO
1E-3	M (except for OHM and HZ)*	MILLI
1E-6	U	MICRO
1E-9	N	NANO
1E-12	P	PICO
1E-15	F	FEMTO
1E-18	A	ATTO

*The suffix units, MHZ and MOHM, are special cases that should not be confused with <suffix multiplier>HZ and <suffix multiplier>OHM.

Command Terminator

A command may be terminated with a <new line> (ASCII character decimal 10), an EOI (End-of-Identify) asserted concurrent with last byte, or an EOI asserted concurrent with a <new line> as the last byte.

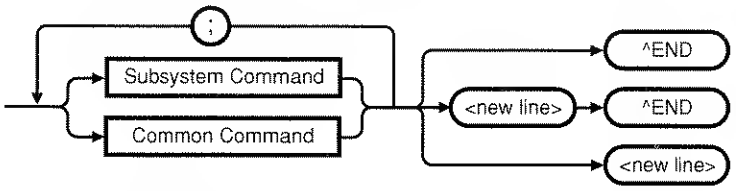
Using Multiple Commands

Program Messages

Program Messages are a combination of one or more properly formatted SCPI Commands. Program messages always go from a computer to the Counter. They are sent to the Counter over the Counter's HP-IB or serial interface as a sequence of ASCII data messages.

Program Message Syntax

Figure 3-8 shows the simplified syntax of a program message. You can see Common Commands and Subsystem Commands in the same program message. If you send more than one command in one message, you must separate adjacent commands with a semicolon.



NOTE

<new line> = ASCII character decimal 10
^END = EOI asserted concurrent with last byte

Figure 3-8. Simplified Program Message Syntax Diagram

When using IEEE 488.2 Common Commands with SCPI Subsystem commands on the same line, use a semicolon between adjacent commands. For example:

```
*RST;:SENS:AVER ON
```

When multiple subsystem commands are sent in one program message, the first command is always referenced to the root node. Subsequent commands, separated by ";", are referenced to the same level as the preceding command if no ":" is present immediately after the command separator (the semicolon).

For example, sending :SENS:AVER:COUN 5; STAT ON is equivalent to

sending:

```
:SENS:AVER:COUN 5
:SENS:AVER:STAT ON
      or
:SENS:AVER:COUN 5;:SENS:AVER:STAT ON
```

The ":" must be present to distinguish another root level command. For example:

```
:SENS:AVER:COUN 5;:INIT:CONT OFF
```

is equivalent to sending:

```
:SENS:AVER:COUN 5
:INIT:CONT OFF
```

If the ":"(which is following the ";" and is in front of INIT) is omitted, the Counter assumes that the second command is ":SENS:AVER:INIT:CONT OFF" and generates a syntax error.

Overview of Response Message Formats

Response Messages

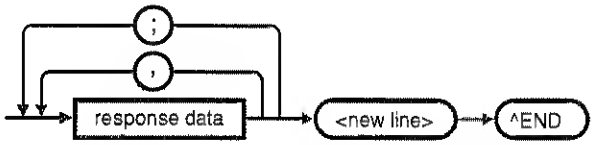
Response messages are data sent from the Counter to a computer in response to a query. (A query is a command followed by a question mark. Queries are used to find out how the Counter is currently configured and to transfer data from the Counter to the computer.)

After receiving a query, the Counter interrogates the requested configuration and places the response in its output queue. The output message remains in the queue until it is read or another command is issued. When read, the message is transmitted across the HP-IB or the serial interface to the computer. You read the message by using some type of enter statement that includes the device address and an appropriate variable. Use a print statement to display the message. The following HP BASIC example illustrates how to query the Counter and display the message:

```
10 OUTPUT 703;":ROSC:SOUR?"
20 ENTER 703; A$
30 PRINT A$
40 END
```


Response Message Syntax

Figure 3-9 shows the simplified syntax of a Response Message. Response messages may contain both commas and semicolon separators. When a single query command returns multiple values, a comma is used to separate each item. When multiple queries are sent in the same program message, the groups of data corresponding to each query are separated by a semicolon. Note that a <new line> ^END is always sent as a response message terminator.



NOTE

<new line> = ASCII character decimal 10
^END = EOI asserted concurrent with last byte
; = multiple response separator (ASCII character decimal 59)
, = data separator within a response (ASCII character decimal 44)

Figure 3-9. Simplified Response Message Syntax Diagram

Response Message Data Types

Table 3-3 contains explanations of response data types.

Table 3-3. Response Message Data Types

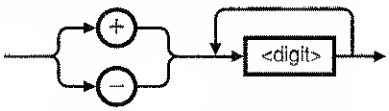
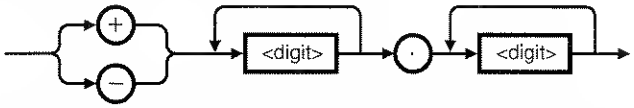
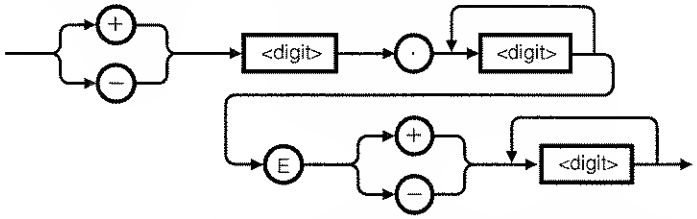
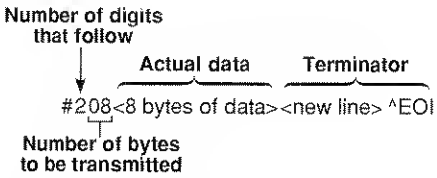
Type	Description
<NR1>	This numeric representation has an implicit radix point.  The maximum number of characters in <NR1> response data is 17 (maximum 16 digits, 1 sign).
<NR2>	This numeric representation has an explicit radix point.  The maximum number of characters in <NR2> response data is 17 (maximum 15 mantissa digits, 1 sign, 1 decimal point).
<NR3>	This numeric representation has an explicit radix point and an exponent.  The maximum number of characters in <NR3> response data is 22 (maximum 15 mantissa digits, 2 signs, 1 decimal point, 1 'E' character, 3 exponent digits).
Not a Number	"Not a Number" is represented by the value 9.91E37. (Not a Number is defined in IEEE 754). The Counter responds with this numeric value when queried for a floating point number it cannot provide. This value will be formatted as an <NR3>.
<Boolean>	A single ASCII-encoded byte, 0 or 1, is returned for the query of settings that use <Boolean> parameters.
<literal>	ASCII-encoded bytes corresponding to the short form of the literal used as the command parameter.

Table 3-3. Response Message Data Types (Continued)

Type	Description
<string>	A string response consists of ASCII characters enclosed by double quotes. For example, string data is used for the "<error description>" portion of :SYST:ERR? response and for [:SENS]:FUNC? response.
<definite length block>	The syntax is a pound sign (#) followed by a non-zero digit representing the number of digits in the subsequent decimal integer. The decimal integer specifies the number of 8-bit data bytes being sent. This is followed by the actual data. The terminator is a line feed asserted with EOI. For example, for transmitting 8 bytes of data, the format might be:  The "2" indicates the number of digits that follow and the two digits "08" indicate the number of *data* bytes to be transmitted. A zero-length block has the format: #0<new line>^EOI <new line> is defined as a single ASCII-encoded byte corresponding to 10 decimal.

Status Reporting

The HP 53150A, 53151A, and 53152A status registers conform to the SCPI and IEEE 488.2 standards.

Figure 3-10 shows all of the status-register groups and queues in the Counter. This is a high level diagram that does not show all the registers that are contained in each group. It is intended as a guide to the bits used in each of these register groups to monitor the Counter's status. Note that a summary of the Standard Status Structure Registers (defined by IEEE 488.2-1987) is shown in addition to the Operation Status and Questionable Data/Signal Register groups.

Refer to the section in this chapter titled "Programming the Counter for Status Reporting" and the flowchart in Figure 3-14 for detailed information on programming the status-reporting system.

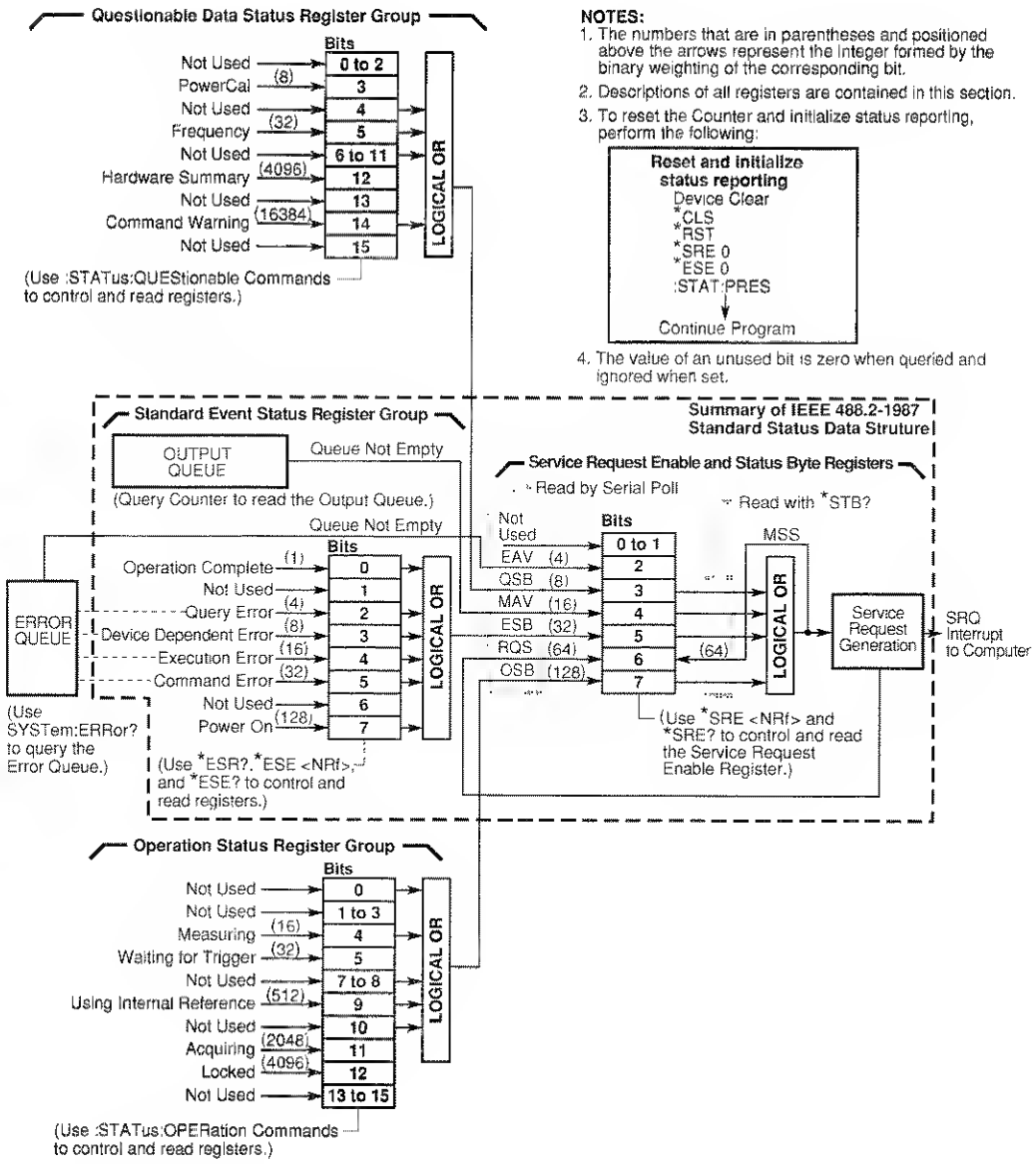


Figure 3-10. 53150A/151A/152A SCPI Status Reporting Summary Functional Diagram

Status Byte Register and Service Request Enable Register

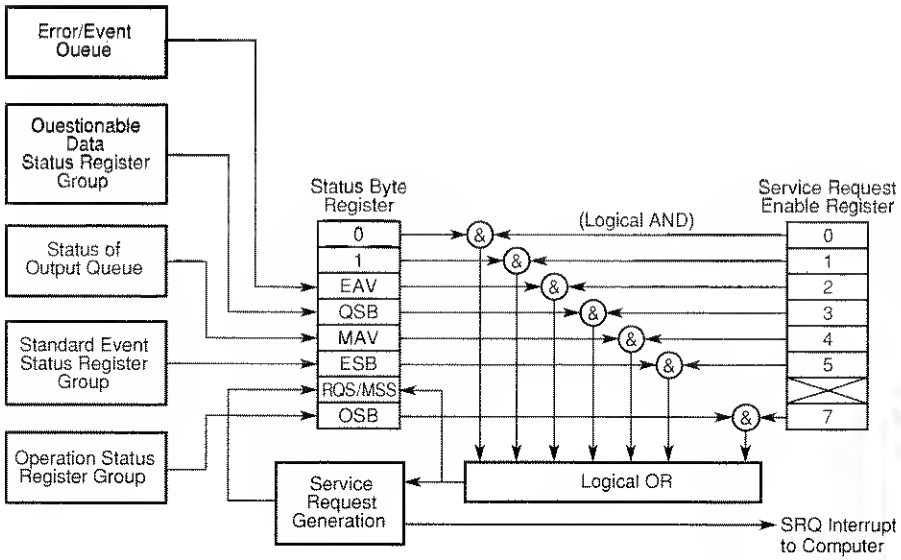


Figure 3-11. Status Byte and Service Request Enable

Status Byte Register

The Status Byte Register is the summary-level register in the status reporting structure. It contains summary bits that monitor activity in the other status registers and queues as shown in Figure 3-11. The Status Byte Register is a live register—its summary bits are set TRUE or FALSE (one or zero) by the presence or absence of the condition which is being summarized.

The Status Byte Register can be read with either a serial poll or the *STB? query, but it is altered only when the state of the overlying status data structures is altered. The entire Status Byte Register can be cleared by sending just the *CLS command to the Counter in a program message.

Table 3-4 lists the Status Byte Register bits and briefly describes eacb bit.

Table 3-4. Status Byte Register

BIT	WEIGHT	SYMBOL	DESCRIPTION
0	—	—	Not used
1	—	—	Not used
2	4	EAV	Error/Event Queue Not Empty
3	8	QSB	Questionable Data/Signal Status Register Summary Bit
4	16	MAV	Message Available Summary Bit
5	32	ESB	Standard Event Status Register Summary Bit
6	64	RQS/MSS	Request Service/Master Status Summary Bit
7	128	OSB	Operation Status Register Summary Bit

A detailed description of each bit in the Status Byte Register follows:

- **Bits 0 - 1** are not used.
- **Bit 2 (EAV)** Summarizes the Error/Event Queue.

  This bit is set when the Error/Event Queue is not empty.
- **Bit 3 (QSB)** summarizes the Questionable Data Status Event Register.

  This bit indicates whether or not one or more of the enabled Questionable Data events have occurred since the last reading or clearing of the Questionable Data Status Event Register.

  This bit is set TRUE (one) when an enabled event in the Questionable Data Status Event Register is set TRUE. Conversely, this bit is set FALSE (zero) when no enabled events are set TRUE.
- **Bit 4 (MAV)** (Message AVailable) summarizes the Output Queue.

  This bit indicates whether or not the Output Queue is empty.

  This bit is set TRUE (one) when the Counter is ready to accept a request by the external computer to output data bytes; that is, the

Output Queue is not empty. This bit is set FALSE (zero) when the Output Queue is empty.

- **Bit 5 (ESB)** summarizes the Standard Event Status Register.

  This bit indicates whether or not one of the enabled Standard Event Status Register events have occurred since the last reading or clearing of the Standard Event Status Register.

  This bit is set TRUE (one) when an enabled event in the Standard Event Status Register is set TRUE. Conversely, this bit is set FALSE (zero) when no enabled events are set TRUE.

- **Bit 6 (RQS/MSS)** summarizes IEEE 488.1 RQS and Master Summary Status.

  When a serial poll is used to read the Status Byte Register, the RQS bit indicates if the device was sending SRQ TRUE. The RQS bit is set FALSE by a serial poll.

  When *STB? is used to read the Status Byte Register, the MSS bit indicates the Master Summary Status. The MSS bit indicates whether or not the Counter has at least one reason for requesting service.

- **Bit 7 (OSB)** summarizes the Operation Status Event Register.

  This bit indicates whether or not one or more of the enabled Operation events have occurred since the last reading or clearing of the Operation Status Event Register.

  This bit is set TRUE (one) when an enabled event in the Operation Status Event Register is set TRUE. Conversely, this bit is set FALSE (zero) when no enabled events are set TRUE.

Service Request Enable Register

The Service Request Enable Register selects which summary bits in the Status Byte Register may cause service requests as shown in Figure 3-7.

Use *SRE to write to this register and *SRE? to read this register.

Use *SRE 0 to clear the register. A cleared register does not allow status information to generate the service requests. (Power-on also clears this register.)

Standard Event Status Register Group

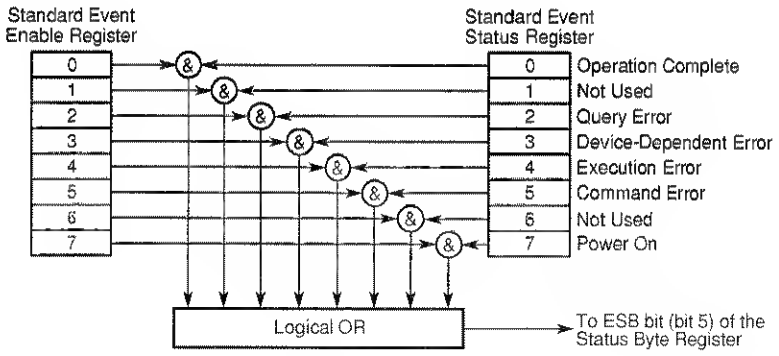


Figure 3-12. Standard Event Status Reporting

Standard Event Status Register

The Standard Event Status Register contains hits that monitor specific IEEE 488.2-defined events as shown in Figure 3-12.

Use *ESR? to read this register.

Use *ESR? or *CLS to clear this register.

Table lists the Standard Event Status Register bits and briefly describes each bit.

Table 3-5. Standard Event Status Register

BIT	WEIGHT	SYMBOL	DESCRIPTION
0	1	OPC	Operation Complete
1	—	(ROC)	Not used because this instrument cannot request permission to become active IEEE 488.1 controller-in-charge.
2	4	QYE	Query Error
3	8	DDE	Device-Specific Error
4	16	EXE	Execution Error
5	32	CME	Command Error
6	—	(URQ)	Not used, because this instrument does not define any local controls as "User Request" controls.
7	128	PON	Power On

A detailed description of each bit in the Standard Event Status Register follows:

- **Bit 0 (Operation Complete)** is an event bit which is generated in response to the *OPC command. This bit indicates that the Counter has completed all pending operations (the pending operation condition has transitioned from TRUE to FALSE).

  If AVERage:STATe is OFF, the command INIT;*OPC sets the OPC bit once the instrument completes a measurement; if AVERage:STATe is ON, the command INIT;*OPC sets the OPC bit once the instrument completes a measurement consisting of AVERage:COUNt measurements.

NOTE

The OPC bit is not in any way affected by the *OPC? query.

- **Bit 1** is not used.
- **Bit 2 (Query Error)** is an event bit which indicates that either 1) an attempt was made to read the Output Queue when it was empty or 2) data in the Output Queue has been lost.

Errors -400 through -499 are query errors.

- **Bit 3 (Device-Specific Error)** is an event bit which indicates an operation did not properly complete due to some condition of the Counter.

  Errors -300 through -399 and all those with positive error numbers are device-specific errors.

- **Bit 4 (Execution Error)** is an event bit which indicates that a command could not be executed 1) because the parameter was out of range or inconsistent with the Counter's capabilities, or 2) because of some condition of the Counter.

  Errors -200 through -299 are execution errors.

- **Bit 5 (Command Error)** is an event bit which indicates one of the following has occurred: 1) an IEEE 488.2 syntax error, 2) a semantic error indicating an unrecognized command, or 3) a Group Execute Trigger was entered into the input buffer inside of a program message.
- **Bit 6** is not used.
- **Bit 7 (Power On)** is an event bit which indicates that an off-to-on transition has occurred in the Counter's power supply.

Standard Event Status Enable Register

The Standard Event Status Enable Register selects which events in the Standard Event Status Register are reflected in the ESB summary bit (bit 5) of the Status Byte Register as shown in Figure 3-8.

Use *ESE to write to this register and *ESE? to read this register.

Use *ESE 0 to clear the register. (Power-on also clears this register.)

The Operation and Questionable Data Status Register Groups

The Operation and Questionable Data Status Register Groups have the following registers:

- a condition register
- one or more transition filters
- an event register
- an event enable register

Figure 3-13 shows the model that these register groups follow.

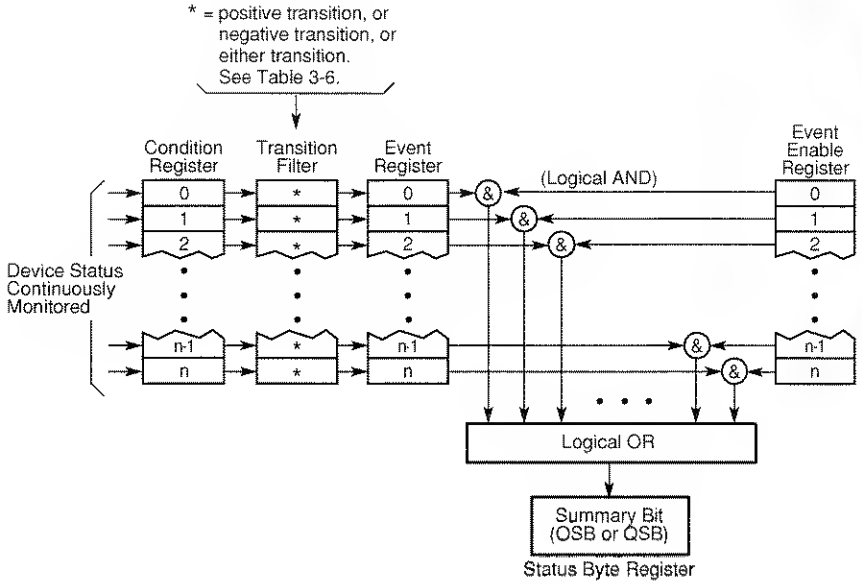


Figure 3-13. Operation and Questionable Data Status Reporting Model

Condition Register

A condition register continuously monitors the hardware and firmware status of the Counter. There is no latching or buffering for this register; it is updated in real time. Reading a condition register does not change its contents.

To read the condition registers use:

:STATus:OPERation:CONDition?

:STATus:QUEStionable:CONDition?

Transition Filter

A transition filter specifies the transition criteria for setting event bits TRUE.

When tbe transition filter specifies a positive transition, the event becomes TRUE when its associated condition makes a FALSE to TRUE transition only.

When the transition filter specifies a negative transition, the event becomes TRUE when its associated condition makes a TRUE to FALSE transition only.

When the transition filters specify both a positive and a negative transition, the event becomes TRUE when its associated condition makes either a FALSE to TRUE or a TRUE to FALSE transition.

A transition filter is defined by positive and negative transition filter registers. Table 3-6 describes how the transition filter registers define the transition criteria for setting an event bit TRUE.

Table 3-6. Transition Filter Definition

Positive Transition Filter Bit	Negative Transition Filter Bit	Transition Which Causes the Event-Bit to be set TRUE
TRUE	FALSE	positive transition
FALSE	TRUE	negative transition
TRUE	TRUE	either a positive or negative transition
FALSE	FALSE	neither transition (event reporting is disabled)

Transition filters are unaffected by *CLS or queries. Transition filters are set to default values by :STATus:PRESet and power-on.

To write to the Operation Status transition filter registers use:

:STATus:OPERation:PTRansition
:STATus:OPERation:NTRansition

To read these registers use:

:STATus:OPERation:PTRansition?
:STATus:OPERation:NTRansition?

Event Register

An event register captures changes in conditions.

An event register hit (event bit) is set TRUE when an associated event occurs. These bits, once set, are "sticky." That is, they cannot be cleared even if they do not reflect the current status of a related condition, until they are read.

To read the event registers use:

```
:STATus:OPERation[:EVENt]?
:STATus:QUEStionable[:EVENt]?
```

Use event register queries or *CLS to clear event registers.

Event Enable Register

An event enable register selects which event bits in the corresponding event register can generate a summary bit.

To write the event enable registers use:

```
:STATus:OPERation:ENABle
:STATus:QUEStionable:ENABle
```

To read tbe event enable registers use:

```
:STATus:OPERation:ENABle?
:STATus:QUEStionable:ENABle?
```

The event enable registers are cleared by :STATus:PRESet and power-on.

Operation Status Register Group

The Operation Status Register Group monitors conditions which are part of the Counter's normal operation and has a complete set of registers that consist of the following:

- a condition register
- a positive transition filter register (PTR)
- a negative transition filter register (NTR)
- an event register
- an event enable register

Table 3-7 lists the Operation Status Register bits and briefly describes each bit. Figure 3-13 shows the model that these register groups follow.

Table 3-7. Operation Status Register

BIT	WEIGHT	DESCRIPTION
0	—	Not used
1 - 3	—	Not used
4	16	Measuring
5	32	Waiting for Trigger
6 - 8	—	Not used
9	512	Using Internal Reference
10	—	Not used
11		Acquiring
12		Locked
13 - 14	—	Not used
15	—	Not used, since some controllers may have difficulty reading a 16-bit unsigned integer. The value of this bit is always 0.

A detailed description of each bit in the Operation Status Register follows:

- **Bits 0-3** are not used.
- **Bit 4 (Measuring)** is a condition bit which indicates the Counter is actively measuring.

  The condition bit is TRUE (one) during a measurement and FALSE (zero) otherwise.
- **Bits 5 (Waiting for Trigger)**

  The condition bit is TRUE (one) when the Counter is in the HOLD mode (INIT:CONT OFF), and it has not been triggered.
- **Bits 6-8** are not used.
- **Bit 9 (Using Internal Reference)** is a condition bit which indicates the Counter is using the internal reference.

  The condition bit is TRUE (one) when the Counter is using the internal reference. The condition bit is FALSE (zero) while the Counter is using the external reference. The setting of this bit is not affected by the setting of the REF OSC option in the user settings menu.
- **Bit 10** is not used.
- **Bit 11 (Acquiring)** indicates that the counter is searching for a signal.

  The condition bit is high while the counter is searching. It goes low when a search is complete, whether or not a signal was found. If a signal was not found, tbe bit goes high again after a delay (when the next search begins).
- **Bit 12 (Locked)** a value of 1 indicates that the counter has found a measurable signal and has locked onto it.
- **Bits 13-15** are not used.

Questionable Data Status Register Group

The Questionable Data Status Register Group monitors SCPI-defined conditions.

NOTE

For this register group, the transition filter is fixed as PTR with all bits set to ones. This cannot be changed or queried.

Table lists the Questionable Data Status Register bits and briefly describes each bit.

Table 3-8. Questionable Data Status Register

BIT	WEIGHT	DESCRIPTION
0 - 2	—	Not used
3	8	Power
4	—	Not used
5	32	Frequency
6 - 11	—	Not used
12	4096	Hardware Summary
13	—	Not Used
14	16384	Command Warning
15	—	Not used, since some controllers may have difficulty reading a 16-bit unsigned integer. The value of this bit is always 0.

A detailed description of each bit in the Questionable Data Status Register Group follows:

- **Bits 0-2** are not used.
- **Bit 3 (Power).** Power Cal is turned off or the Power Cal tables in EEPROM are defective or missing.
- **Bit 5 (Frequency)** is a condition bit which indicates that frequency measurements may be affected by component failures.
- **Bits 6-11** are not used.

- **Bit 12 (Hardware Summary)** This condition is TRUE when an internal hardware fault has been detected, either in normal operation or by the self test.
- **Bit 13** is not used.
- **Bit 14 (Command Warning)** is an *event* bit indicating a command, such as CONFigure or MEASure, ignored a parameter during execution.

  Since this is an event bit, the transition filters have no effect on it.
- **Bit 15** is not used.

Programming the Counter for Status Reporting

Determining the Condition of the Counter

The Counter has status registers that are used to indicate its condition. There are four register groups that can be examined individually, or used to alert a computer. These registers, shown in Figure , are:

- Operation Status Register Group
- Questionable Data/Signal Register Group
- Standard Event Status Register Group
- Status Byte Register Group

The first three groups all have event registers that can be fed into the Status Byte Register. The Status Byte Register can be used to assert the SRQ line and thus alert the computer that the Counter needs attention. The following examples show how each of the register groups can be used. (Figure 3-14 is a flowchart that shows how to program the Counter for Status Reporting.)

Resetting the Counter and Clearing the Remote Interface—Example 1

Before attempting any programming, it is a good idea to set the Counter to a known state. The following command grouping shows how to reset the Counter. Before issuing these commands, execute a device clear to reset the interface and Counter. Consult your interface card's documentation for how to issue a device clear since the device clear command will be specific to the interface you are using. Perform the following:

1. Issue an Interface Clear and a Device Clear. (See your computer or interface card documentation on how to issue this command).
2. Issue the following commands:

```
*RST
*CLS
*SRE 0
*ESE 0
:STAT:PRES
```

Using the Standard Event Status Register to Trap an Incorrect Command—Example 2

The following command grouping shows how to use the Standard Event Status Register and the Status Byte Register to alert the computer when an incorrect command is sent to the Counter. The command *ESE 32 tells the Counter to summarize the command error bit (bit 5 of the Event Status Register) in the Status Byte Register. The command error bit is set when an incorrect command is received by the Counter. The command *SRE 32 tells the Counter to assert the SRQ line when the Event Status Register summary bit is set to 1. If the Counter is serial-polled after a command error, the serial poll result is 96 (Bit 6 + Bit 5).

Event Status Register

*ESE 32 Enable for bad command.

*SRE 32 Assert SRQ from Standard Event Status Register summary.

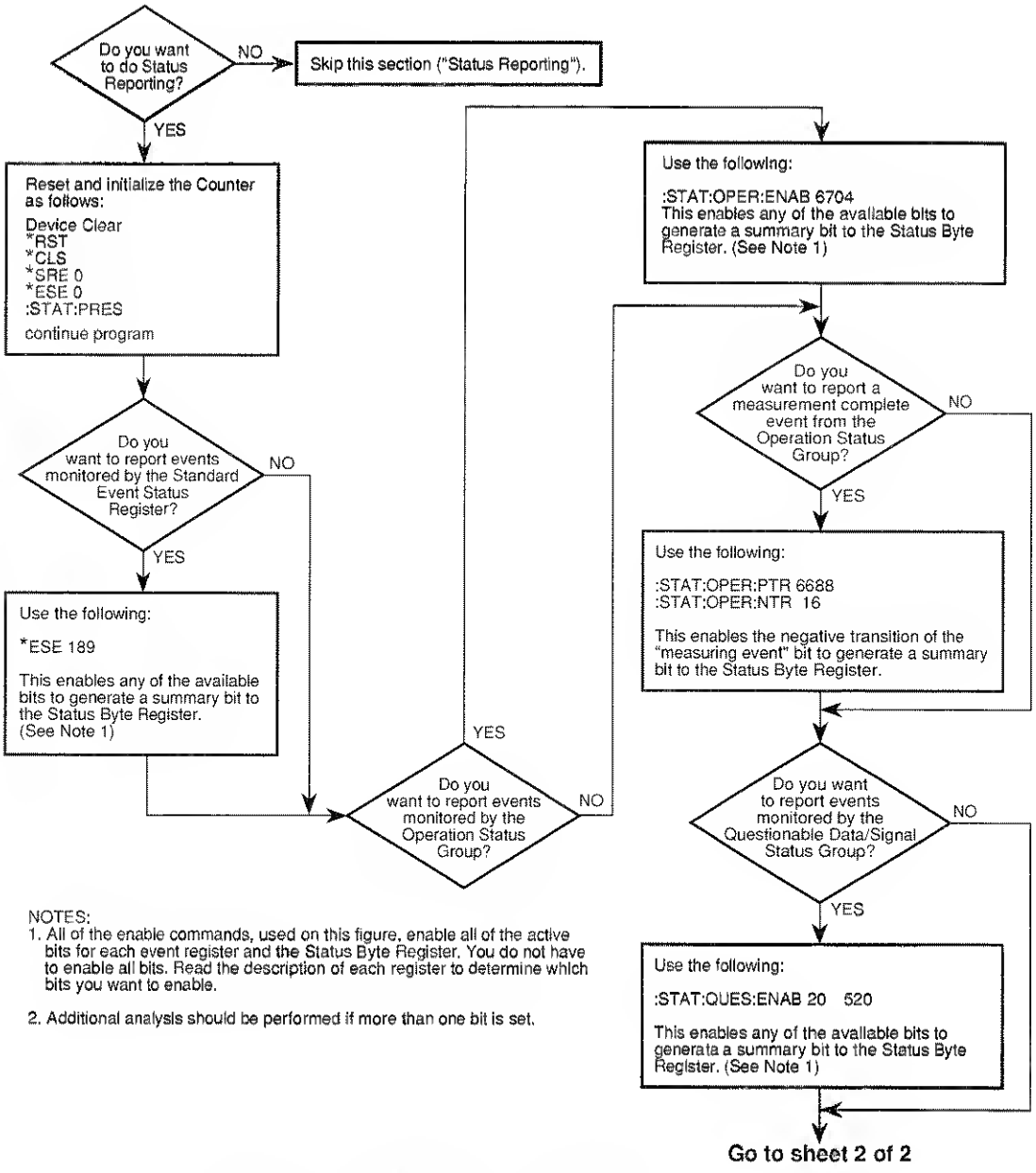
Using the Operation Status Register to Alert the Computer When Measuring has Completed—Example 3

The following command grouping illustrates how to use the Operation Status register and the Status Byte register to alert the computer when measuring has completed. This is useful if the Counter is making a long measurement. When the measurement is complete, the Counter can alert the computer.

The first line tells the Counter to watch for a negative transition from true (measuring) to false (non-measuring) of bit 4. This negative transition indicates that the Counter has completed a measurement. The next line tells the Counter to summarize the detected event (bit 4 of the Operation Status Register) in the Status Byte Register. The command *SRE 128 tells the Counter to assert SRQ when the summary bit for the Operation Status register is set to 1. A serial poll will return 192 when a measurement has completed.

Operation Status Register

:STAT:OPER:PTR 0; NTR 16	Detect transition from measuring to non-measuring.
:STAT:OPER:ENABLE 16	Enable to detect measuring.
*SRE 128	Assert SRQ on Operation Summary bit.



NOTES:
1. All of the enable commands, used on this figure, enable all of the active bits for each event register and the Status Byte Register. You do not have to enable all bits. Read the description of each register to determine which bits you want to enable.

2. Additional analysis should be performed if more than one bit is set.

Figure 3-14. Status Reporting Flowchart (1 of 2)

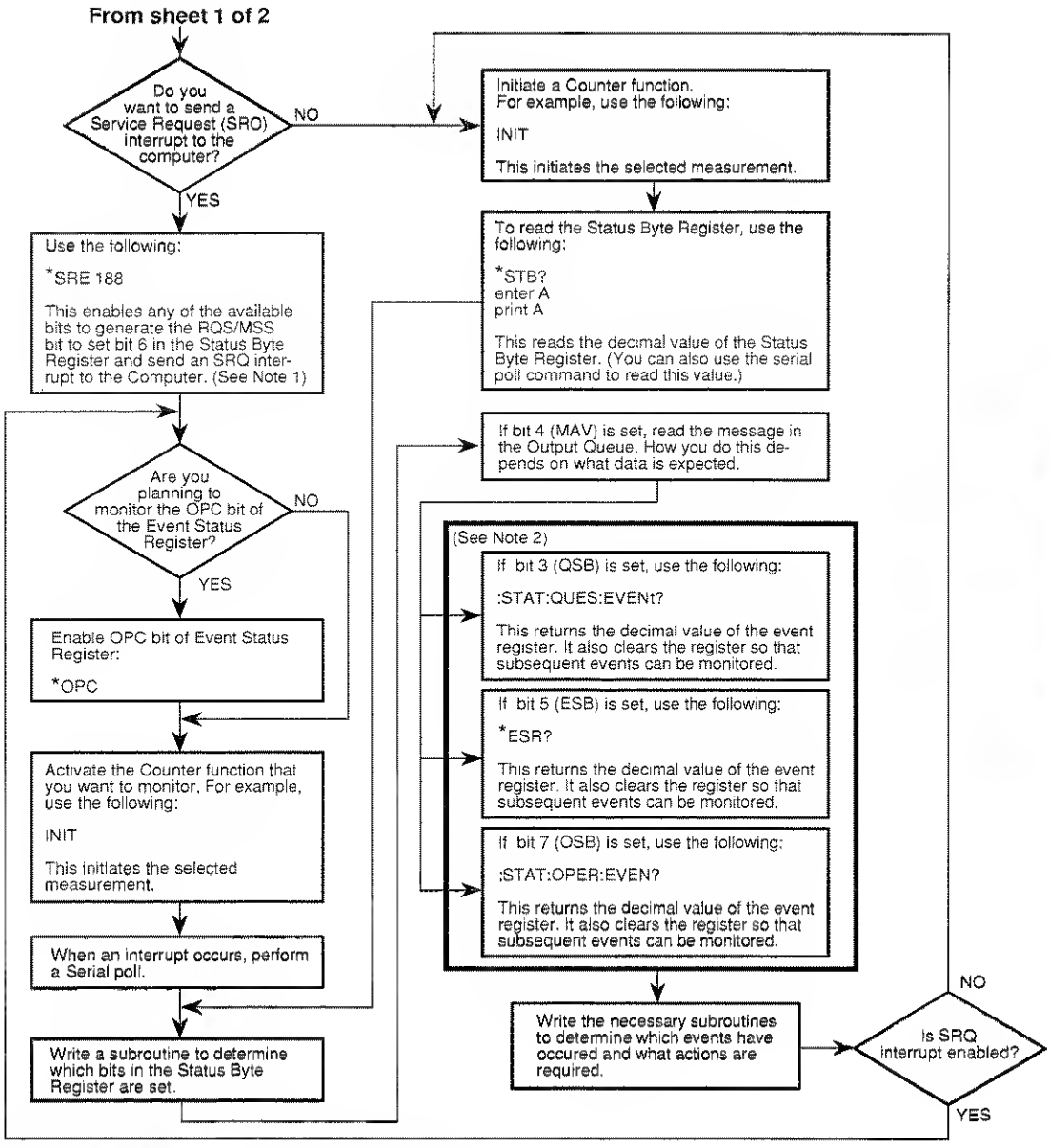


Figure 3-14. Status Reporting Flowchart (2 of 2)

Programming the Counter to Display Results

Configuring the Counter's Display

The Counter has three display modes:

1. Raw results. This display mode is used on power-up.
2. Relative results - results modified by offset values
3. Display Disabled - All LCD display segments disabled.

The following command groupings show how to program the Counter to any of the above display modes.

Commands for Displaying Results

Command for Displaying Raw Results

The following command always causes raw measurement results to be displayed:

*RST	Turns off all offsets.

Commands for Displaying Relative Results

FREQ:OFFS:STAT ON	Turns on offsets.
POW:AC:REF:STAT ON	Turns on offsets.

Commands for Enabling and Disabling the Display

The Counter display can be turned on or off. The normal condition is for the display to be on.

:DISP:ENABLE OFF	Disable the display, all segments off.
:DISP:ENABLE ON	Normal display mode.

Programming the Counter to Synchronize Measurements

Synchronizing Measurement Completion

There are three commands for synchronizing the end of a measurement and computer transfer of data:

1. The *WAI command
2. The *OPC? command
3. The *OPC command to assert SRQ

The following discussion shows how to use these three commands.

Resetting the Counter and Clearing the Interface

Before attempting any programming, it is best to set the Counter to a known state. The following command grouping illustrates how to reset the Counter. Before issuing these commands, execute a device clear to reset the interface and the Counter. You should consult your interface card's documentation for information on issuing a device clear, since the device clear command is specific to the interface you are using. Perform the following steps:

1. Issue an Interface Clear and a Device Clear. (See your computer or interface card documentation for information on how to issue this command.)
2. Issue the following commands:

```
*RST
*CLS
*SRE 0
*ESE 0
:STAT:PRES
```

Using the *WAI Command

This command is most useful when only the Counter is on the bus, and you want the Counter to send the data when it is ready. In this example, the Counter is instructed to take 50 measurements and return the average for these 50 measurements. The *WAI command that follows the :INIT command instructs the Counter to hold off execution of any further commands until the 50 measurements are complete. When the Counter has completed the 50 measurements and averages, it executes the DATA? command, which requests the results.

:AVERAGE:STATE ON	Enable averaging.
:AVERAGE:COUNT 50	Base the result on 50 measurements
:INIT	Start 50 measurements.
*WAI	Wait until 50 measurements are complete before parsing another command. At this point, commands can he issued to other instruments. The Counter stores subsequent commands but ignores them until the measurement is complete.
DATA?	Asks for the results of the 50 measurements. This command is not executed until all 50 measurements are complete and the average is computed.

Using the *OPC? Command

This method is useful if you want to hold off execution of the program while you wait for the Counter to complete any pending activity. In the *WAI example above, the line following the *WAI command is accepted by the Counter. However, the Counter does not execute the command

because of the preceding *WAI command. If this line had been a command to address another instrument, it would be immediately executed. If you had wanted to hold off the command to another instrument, you would use the *OPC? command instead of the *WAI command.

:AVERAGE:COUNT 50	On INIT, take 50 measurements.
:AVERAGE ON	Enable averaging.
:INIT	Start making measurements.
*OPC?	Tell Counter to put a 1 in the output buffer when 50th measurement is complete.

Read the Counter. The program waits until the Counter returns a 1. (The HP-IB timeout must he set so that it is longer than the expected measurement time.)

Using the *OPC Command to Assert SRQ

This method is recommended when the Counter is interfaced with many other instruments, any of which can assert SRQ. The commands *OPC, *ESE 1 and *SRE 32 are used to assert the SRQ line to alert the computer that the Counter has completed a measurement. It is up to the computer to use the serial poll command to determine which of the instruments on the bus requested service.

Of the three procedures discussed here, this one is the most flexible, but it is also the most complex:

:AVERAGE:COUNT 50	On INIT, take N measurements.
:AVERAGE ON	Enable averaging.
*ESE 1	Summarize OPC hit for Status Byte Register.
*SRE 32	SRQ when event summary bit is 1.

Set up program to specify service routine and enable interrupt when SRQ is asserted:

```
:INIT          Start measurements.
*OPC           Enable OPC bit.
```

The program can do other things while it is waiting for SRQ.

When SRQ occurs, and the Counter has been identified as the cause of the SRQ, ask for the data:

```
DATA?          Ask for data.
```

Writing SCPI Programs

Figure 3-15 is a general summation of how to write SCPI programs. It shows a typical sequence you might go through in the process of writing a program. You do not have to follow this exact sequence, but it will help you to become familiar with the Counter's capahilities and to direct you to sections of the guide which will be useful while writing programs.

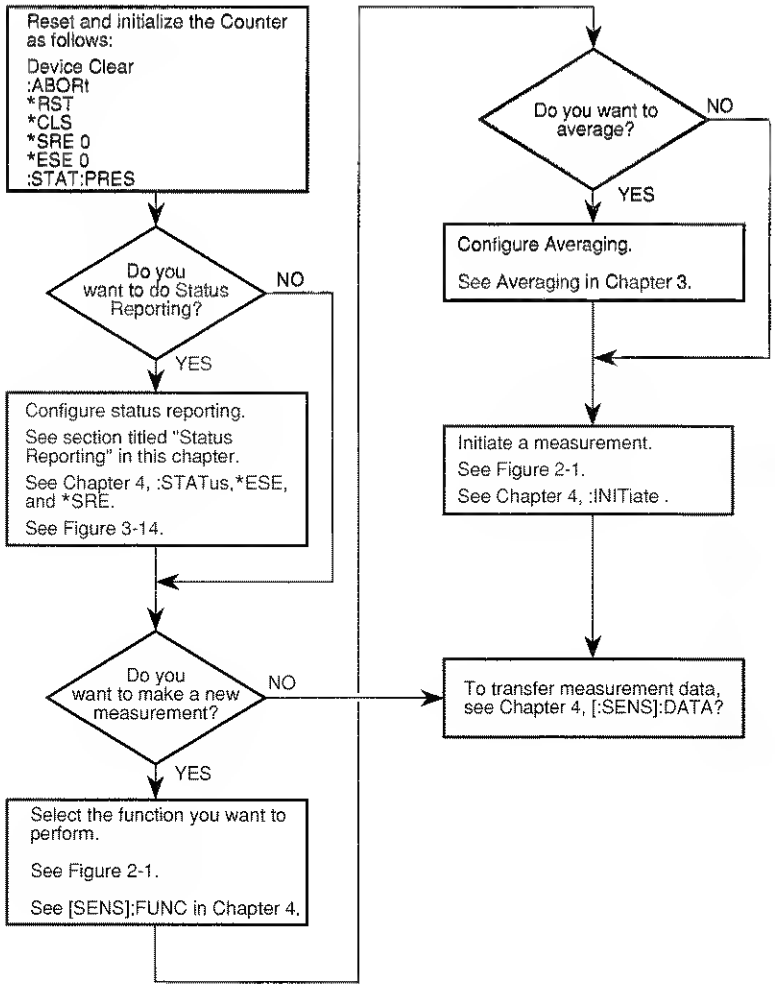


Figure 3-15. SCPI Programming Flowchart

Programming Examples

In this section, you will see how to program the HP 53150A/151A/152A to make common measurements. Examples are provided in the following programming languages:

- HP BASIC
- Microsoft® QuickBASIC
- C

Using HP BASIC

This guide uses doubles quotes to enclose string parameters in syntax descriptions, but uses single quotes in the HP BASIC programming examples for readability.

The Counter allows string parameters to be enclosed by either double or single quotes. Each method is discussed in the following sub-sections.

Sending Double-Quoted and Single-Quoted Strings

In HP BASIC OUTPUT statements, strings enclosed in double quotes need special consideration. For example, you can send the FUNC "FREQ 1" command as follows:

```
OUTPUT 703;"FUNC ""FREQ 1"""
```

Note that a pair of double quotes (shown in bold) is required to embed a double quote within an HP BASIC string. For more readable HP BASIC OUTPUT statements, you can send the following command instead:

```
OUTPUT 703;"FUNC 'FREQ 1'"
```

Note that the pair of single quotes (shown in bold) is more readable.

---

* Microsoft is a U.S. registered trademark of Microsoft Corporation.
** Turbo C is a product of Borland International, Inc.

Using C

The C examples assume you have an HP 82335A HP-IB Interface card inside your IBM PC or compatible.

List of the Programming Examples

The following examples are provided:

1. Making a Frequency Measurement (HP BASIC)
2. Making a Frequency Measurement (QuickBASIC)
3. Making a Frequency Measurement (C)

NOTE

All programming examples use the ASCII format to transfer data from the Counter to the computer. The ASCII format is the default format when *RST is used.

Making a Frequency Measurement (HP BASIC)

```
10 ! This program sets up the counter to make 10 frequency
20 ! measurements on channel 2.
30 ! The results are displayed on the computer CRT.
40 ! ASCII format is used to preserve resolution.
50 !
60    INTEGER I                          ! Declare variables
70    DIM Freq$(10)[22]                   ! Declare string to enter data
80                                        !Using strings to enter ASCII format
90                                        ! data yields results formatted to the
100                                       ! correct resolution. ASCII is the
110                                       ! default format for the counter.
120   Samples=10                          ! Take 10 measurements
130    !
140   ASSIGN @Count TO 703                ! Assign I/O path for counter
150   CLEAR 703                           ! Clear the counter and interface
160   OUTPUT @Count;"*RST"                ! Reset the counter
170   OUTPUT @Count;"*CLS"                ! Clear event registers and error queue
180   OUTPUT @Count;"*SRE 0"              ! Clear service request enable register
190   OUTPUT @Count;"*ESE 0"              ! Clear event status enable register
200   OUTPUT @Count;":STAT:PRES"          ! Preset enable registers and
210                                       ! transition filters for operation and
220                                       ! questionable status structures.
230   OUTPUT @Count;":FUNC 'FREQ 2'"      ! Measure frequency on channel 1
270    !
280   CLEAR SCREEN                        ! Clear the computer display
290   FOR I=1 TO Samples                  ! Start making measurements
300     OUTPUT @Count;"INIT:*WAI;:DATA?" ! Start a measurement and
320     ENTER @Count;Freq$(I)             ! fetch the data
330     PRINT USING "11A,DD,4A,22A,3A";"Frequency (";I;") = ";Freq$(I);" Hz"
340   NEXT I
350   LOCAL 703                           ! Return counter to local
360   END
```

Making a Frequency Measurement (QuickBASIC)

```
'This program configures the counter to make 10 frequency measurements
'on channel 2.
'The results are printed on the computer monitor.
'Data is sent in ASCII format to preseve resolution.
'
'The SUB sendhp sends commands to the counter

DECLARE SUB sendhp (code$)
REM $INCLUDE: 'QBSETUP.BAS'              'Required by HP 82335A
DIM SHARED source AS LONG                'Address and select code
DIM i AS INTEGER                         'i is used for loops
DIM samples AS INTEGER
samples = 10                             'Number of measurements
DIM freqs(10) AS STRING * 23             'String to be read
                                         'Reading ASCII formatted data
                                         'gives results to the correct
                                         'resolution. Must be read into
                                         'a string. The maximum number
                                         'of characters that can ever be
                                         'sent is 20 per measurement.
source& = 703                            'Counter at address 3
isc&p; = 7                               'Select code 7
state% = 1                               'Used in IOEOI

CLS                                      'Clear screen
CALL IOEOI(isc&p;, state%)               'Make sure EOI enabled
CALL IOCLEAR(source&)                    'Clear the counter and interface
CALL sendhp("*RST")                      'Reset counter and stop autotriggering
CALL sendhp("*CLS")                      'Clear event registers and error queue
CALL sendhp("*SRE 0")                    'Clear service request enable register
CALL sendhp("*ESE 0")                    'Clear event status enable register
CALL sendhp(":STAT:PRES")                'Preset enable registers and
transition
                                         'filters for operation and questionable
                                         'status structures
CALL sendhp(":func " + CHR$(34) + "FREQ 2" + CHR$(34))  'Measure frequency
CLS                                      'Clear computer screen
FOR i = 1 TO samples
CALL sendhp("INIT;*WAI;:DATA?")          'Initiate a measurement and
                                         'get the result
CALL IOENTERS(source&ng, freqs(i), 23, actf%)  'Read the ASCII characters
PRINT "Frequency"; i; "= "; freqs(i)
NEXT i
END

'  Subroutine to send command to HP 5315xA/
SUB sendhp (code$)
CALL iooutputs(source, code$, LEN(code$))
END SUB
```

Making a Frequency Measurement (C)

```
/* This program configures the counter to make 10 frequency measurements
   on channel 1 followed by 10 on channel 2.
   The results are displayed on the computer monitor.
   The program comments discuss the meaning of each command.
   ASCII result format is used to preserve resolution. */
#include <stdio.h
#include <string.h
#include "CHPIB.H"
#include "CFUNC.H"
void sendhp(char *);    /* function to send command to counter */
/* global data */
long ctr=703;          /* Counter is at address 03. HP-IB is at select code 7 */
int  error;

void main()
{
    long isc=7;              /* Select code 7                              */
    int state=1;             /* Used in IOEOI                              */
    int i;                   /* Used for loop counter                      */
    int samples=10;          /* Number of measurements to take             */
    int length=33;           /* Max number of bytes per measurements       */
    char freq[33];           /* Array to hold frequency string             */
    IORESET(isc);            /* Clear the HP-IE interface                  */
    sendhp("*RST");          /* Reset the counter                          */
    sendhp("*CLS");          /* Clear event registers and error queue      */
    sendhp("*SRE 0");        /* Clear service request enable register      */
    sendhp("*ESE 0");        /* Clear event status enable register         */
    sendhp(":STAT:PRES");    /* Preset enable registers and transition
                                filters for operation and questionable
                                status structures                          */
    IOEOI(isc,state);        /* Enable use of EOI                          */

    sendhp(":FUNC 'FREQ 1'");/* Measure frequency on channel 1
                                Note that the function must
                                be a quoted string. The actual
                                string sent to the counter is 'FREQ 1'.    */
    for (i=1; i<=samples ;i++)
        {
          sendhp("INIT");                /* Start a measurement           */
          sendhp("DATA?");
          IOENTERS(ctr,freq,&length);    /* fetch the data                */
          length=strlen(freq);           /* Get length of result so       */
          freq[length-1]='\0';           /* the linefeed can be removed*/
          printf ("Frequency %d = %s Hz\n",i,freq);
        }
    printf("Press a key to continue\n");
    getch();
    sendhp(":FUNC 'FREQ 2'");            /* Change channels               */
    for (i=1; i<=samples ;i++)
        {
          sendhp("INIT;*WAI;:DATA?");    /* Start a measurement           */
          IOENTERS(ctr,freq,&length);    /* fetch the data                */
          length=strlen(freq);           /* Get length of result so       */
          freq[length-1]='\0';           /* the linefeed can be removed*/
          printf ("Frequency %d = %s Hz\n",i,freq);
        }
    printf("Press a key to continue\n");
```

```
    getch();
}

/* Function to send command to HP 5315xA */
void sendhp(hpib_cmd)
char *hpib_cmd;
{
   char hpcmd[80];                          /* Variables used by function */
   int length;

   strcpy(hpcmd, hpib_cmd);
   length=strlen(hpcmd);
   error=IOOUTPUTS(ctr, hpcmd, length);    /* Send command to HP 5315xA  */
   if (error!=0)
   printf("Error during HP-IB: %d Command %s\n",error,hpcmd);
}
```

4

Command Reference

A Dictionary

Introduction

This chapter describes the SCPI Subsystem commands and the IEEE 488.2 Common Commands for the HP 53150A/151A/152A Microwave Frequency Counters. The information in this chapter is intended to help you program the Counter over its HP-IB or RS-232 serial interface.

The commands are presented in alphabetical order.

- SCPI Subsystem commands are described on pages 4-4 through 4-43.
- IEEE 488.2 Common command descriptions start on page 4-44.
- A description of the Group Execute Trigger command is also included on page 4-7.

For each command description:

- Where the phrase "Sets or queries" is used, the command setting can he queried by omitting the parameter and appending a "?" to the last command keyword.
- For example,

  :ROSC:SOUR INT | EXT

  can be queried with

  :ROSC:SOUR?
- Unless otherwise noted, a command described as an *event* cannot be queried.
- Unless otherwise noted, the command setting is affected by *SAV/*RCL.
- The square brackets, [ ], are used to indicate that the element(s) within the brackets are optional. Note, the brackets are NOT part of the command and should not be sent to the Counter.
- The vertical bar, |, is used to mean "OR" and is used to separate alternative options.

- The short form of keywords is shown in uppercase.
- Quotation marks may be part of the command's parameter; the quotation marks shown must be sent to the Counter.
- Unless otherwise noted, a command is sequential (not overlapped).

See Chapter 3 of this guide for details regarding command syntax, parameter types, and query response types.

:ABORt Command

COMMAND **:ABORt**

This command causes the Counter to abort, as quickly as possible, any measurement in progress.

The :ABORt command is not complete until the current measurement is stopped. The execution of an ABORt command sets false any Pending Operation Flags that were set true by initiation of measuring.

COMMENTS

- If :ABORt is issued while the measurement cycle is idle (:INIT:CONT OFF and pending operation flag is false), the command is ignored.
- If :ABORt is issued while a single measurement is in progress, the measurement is aborted and the pending operation flag is set false.
- If :ABORt is issued while repetitive measurements are being made (:INIT:CONT ON), the measurement in progress is aborted, and tbe pending operation flag is set false. Then, a new measurement is automatically initiated, and the pending operation flag is set true.
- If :ABORt is issued while a block of measurements, such as an average, is in progress, the measurement block is aborted, and the pending operation flag is set false.
- When a measurement or block of measurements is aborted, the Measuring bit in the Operation Status Register is set false.
- Aborting a measurement in progress invalidates the result.

RELATED FRONT-PANEL KEYS

Reset/Local

:DISPlay Subsystem

This subsystem controls the selection and presentation of textual information on the Counter's display. This information includes measurement results. :DISPlay is independent of, and does not modify, how data is returned to the controller.

See the section titled "Programming the Counter to Display Results" on page 3-46 of this guide.

COMMAND **:DISPlay[:WINDow]:BACKground[:STATe] . . . <Boolean>**

Turns the display backlight ON or OFF.

QUERY RESPONSE
- Single ASCII-encoded byte, 0 or 1.
- A value of 0 indicates OFF; a value of 1 indicates ON.

COMMENTS
- *RST: ON
- Turning the display backlight OFF conserves battery power (if the Counter has the Battery option).
- If the Counter is operating from battery power (Battery option required), the display backlight is automatically turned off after a period of inactivity. When activity is subsequently detected (a measurable signal is applied, a front-panel key is pressed, or a setting is changed by a command sent over the HP-IB or the RS-232 interface), the display backlight is restored to its previous state (i.e., if the backlight was set to OFF, it remains off; if the backlight was set to ON, it is turned on).

RELATED FRONT-PANEL KEYS
- **On/Off** (**Shift** + **Clear**)

COMMAND **:DISPlay:ENABle . . . <Boolean>**

Sets or queries whether the entire display (annunciators and indicators, with the exception of Rmt) is visible.

QUERY RESPONSE
- Single ASCII-encoded byte, 0 or 1.
- A value of 0 indicates OFF; a value of 1 indicates ON.

COMMENTS
- *RST: ON
- This value is unaffected by *SAV/*RCL.

RELATED FRONT-PANEL KEYS None

Group Execute Trigger (GET)

COMMAND **GET**

The full capability of the Group Execute Trigger IEEE 488.1 interface function is implemented in the Counter. This function permits the Counter to have its operation initiated over the Bus. In response to the IEEE 488.1 Group Execute Trigger (GET) remote interface message (while the Counter is addressed to listen), the Counter performs the action defined by the *DDT command (see page 4-46).

RELATED FRONT-PANEL KEYS None

:INITiate Subsystem

This subsystem controls the initiation of a measurement.

COMMAND **:INITiate:CONTinuous . . . <Boolean>**

Sets or queries the state of continuously initiated measurements.

When CONTinuous is set to OFF, no measurements are made until CONTinuous is set to ON or :INITiate[:IMMediate] is received. Once CONTinuous is set to ON, a new measurement is initiated. On the completion of each measurement, with CONTinuous ON, another measurement immediately commences.

QUERY RESPONSE

- Single ASCII-encoded byte, 0 or 1.
- A value of 0 indicates OFF; a value of 1 indicates ON.

COMMENTS

- *RST: OFF
- The commencement of the first measurement due to setting :INITiate:CONTinuous to ON sets the Pending Operation Flag to true. The Pending Operation Flag is set false by aborting a measurement, or by the completion of the last measurement after :INITiate:CONTinuous is set OFF.
- With the measurements being made continuously, the :ABORt command aborts the current measurement in progress, however, the value of :INITiate:CONTinuous is unaffected. If CONTinuous was set to ON prior to receiving :ABORt, it remains ON and a new measurement begins.
- When a single measurement is in progress (:INIT:CONT is OFF):
  - Error -213 (Init ignored) is generated and the state of INIT:CONT is unaffected by :INIT:CONT ON.
  - Error -210 (Trigger error) is generated by INIT:CONT OFF.

NOTE

The Counter powers up with :INIT:CONT set to ON, but *RST sets :INIT:CONT to OFF.

COMMAND **:INITiate[:IMMediate]**

This event command causes the instrument to initiate either a single measurement or a block of measurements.

COMMENT

This command is an overlapped command (see IEEE 488.2, Section 12). Beginning a measurement or block of measurements with an :INITiate[:IMMediate] sets the Pending Operation Flag to true. Completing the measurement or block of measurements (normally or by aborting) sets Pending Operation Flag to false.

RELATED FRONT-PANEL KEYS

None

:INPut Subsystem

This subsystem controls the characteristics of the Counter's input ports.

COMMAND **:INPut:FILTer:[:LPASs][:STATe] . . . <Boolean>**

Sets or queries the state of the Channel 1 low-pass filter.

QUERY RESPONSE

- Single ASCII-encoded byte, 0 or 1.
- A value of 0 indicates OFF; a value of 1 indicates ON.

COMMENT

*RST: OFF

RELATED FRONT-PANEL KEYS

Shift + **Menu** (CH1 LPF > OFF | ON)

:MEASure Subsystem

The :MEASure subsystem commands allow you to configure the Counter, initiate measurements, and place the results in the Output Queue using a minimum number of commands. These commands are described in detail in this section.

Measurement Instructions (:CONFigure, :FETCh, :MEASure, :READ)

The purpose of these commands is to acquire data using a set of high-level instructions. These commands are structured to allow you to trade off interchangeability between instruments. The :MEASure query provides the ability to configure the instrument, take a measurement, and store the results in the Output Queue in a single operation.

When more precise control of the measurement is required, the :CONFigure and :READ? commands can he used. The :CONFigure command is used to configure the instrument for the measurement to be taken, and the :READ? command acquires the data, performs any required post processing, and then places the results in the Output Queue. This allows you to configure the instrument generically (using :CONFigure) and then to customize the measurement with other commands (for example, commands from the [:SENSe] subsystem). The :READ? command completes the measurement process.

The :READ? command is composed of the :INITiate[:IMMediate] and :FETCh? commands. :INITiate[:IMMediate] performs the data acquisition. :FETCh? performs the post-processing function (if any) and places the result in the Output Queue. This allows more than one FETCh? on a single set of acquired data.

The functions of the measurement instruction commands are summarized in Table 4-1.

Table 4-1. Summary of the Measurement Instruction Commands

Command	Description
:MEASure query	This command is the simplest to use, but it allows little flexibility. This command lets the Counter configure *itself* for an optimal measurement, initiate the measurement, and return the result; i.e., it provides a complete measurement sequence (:MEAS query is equivalent to the :CONF, :INIT, :FETC? command sequence, but with no flexibility.)
:CONFigure :READ?	The combined use of these two commands allows for more control when the Counter performs a measurement, initiates a measurement, and returns the result. Use this command sequence when you want to customize the configuration between the measurement setup and acquisition.
:CONFigure :INITiate :FETCh?	This combination of commands provides the most flexibility the Measure Instructions allow. This command sequence configures the Counter, initiates the measurement as specified, and returns the result.

The <source_list> parameter has the same syntax as SCPI <channel_list>. For example, the Frequency function uses (@1) to specify channel 1.

If the instrument receives an unexpected parameter, it processes the command, ignoring the unexpected parameter, and sets the "Command Warning" bit of the Data Questionable status reporting structure.

The response format for :MEASure query, :READ?, and :FETCh? is ASCII data. If no valid data is available, the Counter generates error -230 (Data corrupt or stale).

COMMAND **:CONFigure[:SCALar]:<function> <parameters> [,<source_list>]**

Configures the instrument to perform the specified function but does not initiate the measurement.

COMMENTS

- Use :INITiate:FETCh? or :READ? to make and query a measurement.
- Parameters (other than <source_list>) can be defaulted by substituting the keyword DEFault. The <source_list> parameter can be defaulted by omitting it. The default values are specified hy the particular function description.
- This command defaults several Counter settings. To cbange the function only, while leaving all otber Counter settings as they are, use [:SENS]:FUNC[:ON].
- If an <expected_value> parameter is outside the measurement capabilities of the Counter model, an error is generated, and the command does not execute.
- For :POWer, <resolution> must be defaulted or set to 0.01[dB].
- See "Descriptions of the Measurement Functions" on page 4-16 for a description of each of the measurement functions.
- See Table 4-2 for a summary of the <function>, <parameters>, and <source_list> for each of the measurement functions.

COMMAND **:CONFigure?**

Queries the function configured by the last :CONFigure command or :MEASure query.

QUERY RESPONSE

- A string of the form: "<function> <parameters>[,<source_list>]." The leading colon is omitted from the <function>.
- The response is unaffected by *RST, recall, and [:SENS]:FUNC.

COMMENTS

- If the instrument state has changed through commands other than :CONFigure or the :MEASure query, the instrument does not track these changes, and the query response does not reflect these changes.
- For :POWer, <resolution> must be defaulted or set to 0.01[dB].

- If an <expected_value> parameter is outside the measurement capabilities of the Counter model, an error is generated, and the command does not execute.
- See "Descriptions of the Measurement Functions" on page 4-16 for a description of each of the measurement functions.
- See Table 4-2 for a summary of the <function>, <parameters>, and <source_list> for each of the measurement functions.

COMMAND **:FETCh?[[:SCALar]:<function>]?**

This query returns the measurement taken by the :INITiate or :READ? command or the :MEASure query.

QUERY RESPONSE If no valid result is available, no result is returned, and error -230 is generated.

COMMENTS
- When [:SCALar]:<function> is specified, the instrument retrieves the specified result if it matches the current measurement type or can be derived from the current measurement type.
- When [:SCALar]:<function> is omitted, the function specified/used by the last :CONFigure, :MEASure, :READ, or FETCh is used, if possible.
- Issuing this query while a measurement is in progress prevents further commands from being processed until the measurement completes. This hold-off action can only be canceled by the completion of the measurement, Device Clear, or power-on.
- If an <expected_value> parameter is outside tbe measurement capabilities of the Counter model, an error is generated, and the command does not execute.
- For :POWer, <resolution> must be defaulted or set to 0.01[dB].
- See "Descriptions of the Measurement Functions" on page 4-16 for a description of each of the measurement functions.
- Refer to Table 4-2 for a summary of the <function>, <parameters>, and <source_list> for each of the measurement functions.

COMMAND **:MEASure[:SCALar]:<function>? <parameters> [,<source_list>]**

This query provides a complete measurement sequence: configuration, measurement initiation, and query for result.

COMMENTS

- This query is used when generic measurement is acceptable, and fine adjustment of Counter settings is not necessary.
- Parameters (other than <source_list>) can be defaulted by substituting the keyword DEFault. The <source_list> parameter can be defaulted by omitting it. The default values are specified by the particular function description.
- For :POWer, <resolution> must be defaulted or set to 0.01[dB].
- If an <expected_value> parameter is outside the measurement capabilities of the Counter model, an error is generated, and the command does not execute.
- Issuing this query while a measurement is in progress aborts the current measurement before initiating the measurement specified in the query. The Counter then waits for the measurement to complete. This has the effect of holding off processing of further commands until the desired measurement completes. This hold-off action can be canceled only by the completion of the measurement, Device Clear, or power-on.
- If an <expected_value> parameter is outside the measurement capabilities of the Counter model, an error is generated, and the command does not execute.
- For :POWer, <resolution> must be defaulted or set to 0.01[dB].
- See "Descriptions of the Measurement Functions" on page 4-16 for a description of each of the measurement functions.
- Refer to Table 4-2 for a summary of the <function>, <parameters>, and <source_list> for each of the measurement functions.

COMMAND **:READ?[[:SCALar]:<function>]?**

This query provides a method for performing a :FETCh? on *fresh* data.

COMMENTS

- This command is commonly used in conjunction with a :CONFigure command to provide a capability similar to :MEASure?, in which the application programmer is allowed to provide fine adjustments to the instrument state by issuing the corresponding commands between :CONFigure and :READ?.
- When [:SCALar]:<function> is specified, the instrument retrieves the specified result if it matches the current measurement type or can be derived from the current measurement type.
- When [:SCALar]:<function> is omitted, the function specified/used by the last :CONFigure, :MEASure, :READ, or FETCh is used, if possible.
- Issuing this query while a measurement is in progress aborts the current measurement and idles the measurement cycle before initiating the desired measurement. The Counter then waits for the measurement to complete. This has the effect of holding off processing of further commands until the desired measurement completes. Tbis hold-off action can be canceled only by the completion of the measurement, Device Clear, or power-on.
- If an <expected_value> parameter is outside the measurement capabilities of the Counter model, an error is generated, and the command does not execute.
- For :POWer, <resolution> must be defaulted or set to 0.01[dB].
- See "Descriptions of the Measurement Functions" on page 4-16 for a description of each of the measurement functions.
- Refer to Table 4-2 for a summary of the <function>, <parameters>, and <source_list> for each of the measurement functions.

Descriptions of the Measurement Functions

Table 4-2 lists the available measurement functions, the parameters that can be used with them, and the valid values for <source_list>.

Table 4-2. The <function>, <parameters>, and <source_list> for the Measure Instruction Commands

<function>	<parameters>	[,<source_list>]*
[:VOLTage]:FREQuency	[<expected_value>[,<resolution>]]	[,(@1) | (@2)]
:POWer[:AC]	[<expected_value>[,<resolution>]]	[,(@2)]

* <source_list> uses the same syntax as SCPI <channel_list>.

COMMAND **:MEASure[:SCALar][:VOLTage]:FREQuency? [<expected_value>[,<resolution>]] [,(@1) | (@2)]**

This command measures frequency.

FUNCTION DESCRIPTION

<u>**<expected_value>**</u>

Channel 1 range 10 Hz to 125 MHz | MIN | MAX | DEF

Channel 2 range: 100 MHz to 20 GHz (53150A), 26.5 GHz (53151A), 46 GHz (53152A) MIN | MAX | DEF

default: 100 MHz

<u>**<resolution>**</u>

description: The value specifies the frequency resolution for the measurement.

values: 1Hz | 10 Hz | 100 Hz | 1 KHz | 10 KHz | 100 KHz | 1 MHz | DEF

default: 1 Hz

<u>**<source_list>**</u>

description: Specifies which front-panel input is used for the measurement.

values: (@1) | (@2)

default: (@2)

<u>**response format:**</u> <NR1>

:MEASure[:SCALar]:POWer[:AC] [<expected_value>[,<resolution>]] [,(@2)]

This command measures power.

FUNCTION DESCRIPTION

<expected_value>

range:	-40 to +10 dBm
default:	0.00
resolution:	0.01

<resolution>

description:	<resolution> is supported only for compatibility with other instruments.
values:	0.01 dB | MIN | MAX | DEF
default:	0.01 dB

<source_list>

description:	Specifies which front-panel input is used for the measurement.
values:	DEF | (@2)
default:	(@2)

response format: <NR2> Power values are returned in dBm.

How to Use the Measurement Instruction Commands

The Measure Instruction commands have a different level of compatibility and flexibility than other commands. The parameters used with commands from the Measure Instruction describe the signal you are going to measure. This means that tbe Measure Instructions give compatibility between instruments since you do not need to know anything about the instrument you are using.

Using :MEAsure

This is the simplest Measurement Instruction command to use, but it does not offer much flexibility. :MEASure causes the Counter to configure itself for a default measurement, starts the measurement, and queries the result. The following example shows how to use the :MEASure query to measure frequency.

Use

```
:MEASURE:FREQ?
```

to execute a default frequency measurement and have the result sent to the controller. The Counter selects settings and carries out the required measurement; it automatically starts the measurement and sends the result to the controller.

You can add parameters to provide more details about the signal you are going to measure.

Use

```
:MEASURE:FREQ?  50 MHZ, 1 HZ
```

where 50 MHz is the expected value (this value can also be sent as 50E6 HZ), and 1Hz is the required resolution.

The channel numbers can also be specified. For example, you can send:

```
:MEASURE:FREQ?  (@1)
:MEASURE:FREQ?  50 MHz, 1 HZ, (@1)
```

Using :CONFigure with :READ?

The :CONFigure command causes the instrument to choose default settings for the specified measurement. :READ? starts the measurement and queries the result.

This sequence operates in the same way as the :MEASure query, but it allows you to insert commands between :CONFigure and :READ? to specify a particular setting.

For example, use

```
:CONF:FREQ  5 GHZ, 1HZ
```

to configure a default frequency measurement, where 1 Hz is the required resolution, and 5 GHz is the expected value.

Use

```
:READ?
```

to start the measurement and query the result.

Using :CONFigure with :INITiate and :FETCh?

The :READ? query is composed of the :INITiate command, which starts the measurement, and the :FETCh? command, which returns the results to the controller.

For example, use

```
:CONF:FREQ  5 GHZ, 1 HZ
```

to configure for a default frequency measurement, where 1 Hz is the required resolution, and 5 GHz is the expected value.

Use

```
:INITIATE
```

to start the measurement.

Use

```
:FETCH?
```

to query for result.

:MEMory Subsystem

This subsystem manages the instrument's memory.

COMMAND **:MEMory:DATA . . . <name>,<data>**

Stores and queries data in the named power-correction profile.

QUERY RESPONSE

- Data points are returned in <definite length arbitrary block> format.
- A data point consists of two to ten comma-separated, NRf format number pairs.

COMMENTS

- Valid profile names: CORR1, CORR2, . . . CORR9
- Valid data is stored in non-volatile memory. Any error in a data block causes all data in that block to be ignored and the data currently in the named profile to be retained.
- Data must be input in the form of two to ten comma-separated, NRf-format number pairs.
- The data-point number pairs in the named profile are automatically sorted by frequency value before the profile is stored.
- When power correction is in use, the Counter uses the data points in the currently selected profile to determine the amount of loss correction to apply.
- If the measured frequency is between two defined data points, the Counter uses the two defined data points to linearly interpolate the appropriate correction value.
- If the measured frequency is above the highest frequency value in the profile (or below the lowest frequency value), the Counter determines a loss-correction value by using the two highest (or lowest) defined data points to extrapolate a linear extension to the curve above the highest data point (or below the lowest data point).

RELATED FRONT-PANEL KEYS **Shift** + **Menu** (PWR CORR)

COMMAND **:MEMory:CLEAR[:NAME] . . . <name>**

Resets the contents of the named power-correction profile to the default configuration.

COMMENTS
- Valid profile names: CORR1, CORR2, . . . CORR9
- The data currently stored in non-volatile memory for the named correction profile is discarded.
- The default power-correction profile configuration consists of two data points, both having loss values of zero. One of these points contains a frequency value of 1 GHz, and the other contains a frequency value equal to the highest frequency the Counter model can measure (53150A: 20 GHz; 53151A: 26.5 GHz; 53152A: 46 GHz). The remaining eight data points contain loss and frequency values of zero.
- This is an event. There is no query form of this command.

RELATED FRONT-PANEL KEYS **Shift + Menu** (PWR CORR)

COMMAND **:MEMory:NSTates?**

Queries the Number of available *SAV/*RCL STates in the instrument.

QUERY RESPONSE
- Numeric data transferred as ASCII bytes in <NR1> format.
- The value returned is 9.
- The response value is one greater than the maximum which can be sent as a parameter to the *SAV and *RCL commands.

COMMENTS Query only.

RELATED FRONT-PANEL KEYS None

[:SENSe] Subsystem

The [:SENSe] subsystem commands are divided into several sections. Each section, or subtree, deals with controls that directly affect instrument-specific settings and not those related to the signal-oriented characteristics.

COMMAND **[:SENSe]:AVERage[STATe] . . . <Boolean>**

Turns averaging ON and OFF. When averaging is ON, each new valid measurement result is the average of the number of measurements specified in the AVERage:COUNt command.

COMMENTS
- An ABORt command interrupts the averaging and prevents a valid measurement result.
- Averaging cannot be turned ON when the value of AVERage:COUNt is one. Attempting to turn averaging ON when AVERage:COUNt=1 causes an error.

RELATED FRONT-PANEL KEYS **Avg**

COMMAND **[:SENSe]:AVERage:COUNt . . . <numeric_value>**

Specifies the number of measurements to combine when AVERage:STATe is ON.

RANGE The acceptable range for the <numeric_value> parameter is 1 to 99.

COMMENT When averaging is ON, some devices may automatically set :COUNt values in the TRIGger subsystem based on the AVERage:COUNt value. This is done to ensure that the TRIGger subsystem provides enough triggers for the average.

RELATED FRONT-PANEL KEYS **Avg**

COMMAND **[:SENSe]:CORRection:CSET:SELect . . . <name>**

Selects a power-correction profile by name from nine available profiles.

Valid profile names: CORR1, CORR2, . . . CORR9

COMMENTS
- *RST: CORR1
- The correction-profile setting is applied for the current session only. To store the profile selection in non-volatile memory, issue a *SAV command.

RELATED FRONT-PANEL KEYS **Shift + Menu** (PWR CORR > OFF | 1 . . . 9)

COMMAND **[:SENSe]:CORRection:CSET:STATe . . . <Boolean>**

Enables or disables power-correction mode with the currently selected power-correction profile applied.

COMMENTS
- *RST: OFF
- The power-correction state is applied for the current session only. To store the setting in non-volatile memory, issue a *SAV command.

RELATED FRONT-PANEL KEYS **Shift + Menu** (PWR CORR > OFF | 1 . . . 9)

COMMAND **[:SENSe]:DATA? . . . [<data handle>]**

Queries the current measurement result data of the :SENSe subsystem.

Valid <data handles>:

"[SENSe:][XNONE:]FREQuency [1] | 2"
"[SENSe:][XNONE:]POWer [2]"

QUERY RESPONSE

- Frequency values are returned in Hz as ASCII bytes in NR1 format.
- Power values are returned in dB in NR2 format.

COMMENTS

- Query only.
- Does not initiate any measurement action.
- The data handle is optional. If it is omitted, the query returns values for all functions enabled by the FUNCtion:ON command.
- Values are returned in the same order as the functions returned by the FUNCtion:ON? query.
- If this query executes while a measurement is in progress, the prior measurement result is returned, if the prior result was not invalidated.

RELATED FRONT-PANEL KEYS

None

COMMAND **[:SENSe]:FILTer:FM:AUTO . . . <Boolean>**

Turns the Counter's ability to automatically compensate for frequency modulation ON or OFF. (*RST: ON)

COMMENTS

- When FM:AUTO is ON, the Counter automatically detects FM signals and modifies its measurement algorithm accordingly.
- Using FM:AUTO increases the time required to compute each measurement but increases the accuracy of FM signal measurements.

RELATED FRONT-PANEL KEYS

Shift + Menu (FM > AUTO | OFF)

COMMAND **[:SENSe]:FREQuency:OFFSet . . . <numeric_value>[<frequency unit>]**

Sets a reference frequency for all other absolute frequency settings in the instrument.

RANGE The acceptable range for the <numeric_value> parameter is 0 to 50 GHz.

UNITS The offset frequency can be specified in Hz, KHz, or MHz only.

COMMENTS
- This command does not affect the hardware settings of the instrument. It affects only the entered and displayed frequencies.
- The Counter accepts only six digits of resolution for an offset entry through this command. For example, the command FREQ:OFFSet 12345.678912MHz results in an offset of approximately 12.3456 GHz.
- The coupling equation for this command is:

  Entered | Displayedfrequency=(Hardwarefrequency)+offset.

RELATED FRONT-PANEL KEYS **Freq Offset**

COMMAND **[:SENSe]:FREQuency:OFFSet:STATe . . . <Boolean>**

When FREQuency:OFFSet is ON, the frequency measurement results are modified by the setting of FREQ:OFFset before being displayed or reported (in response to a query).

RELATED FRONT-PANEL KEYS **Freq Offset**

COMMAND **[:SENSe]:FREQuency:RESolution...<numeric_value>[<frequency_unit>]**

Sets the resolution of the frequency measurement.

The allowable settings for <numeric_value> and <frequency unit> are:

1 Hz, 10 Hz, 100 Hz, 1 KHz, 10 KHz, 100 KHz, and 1 MHz.

COMMENTS
- *RST: 1 Hz
- This command does not affect the unit multipliers of any measurement queries.
- If no unit is specified, the global frequency unit in effect is used.

RELATED FRONT-PANEL KEYS **Resol**

COMMAND **[:SENSe]:FREQuency:TRACking . . . <character_program_data>**

Selects one of three signal-tracking modes (SLOW | FAST | OFF). When TRACking is set to SLOW, the Counter applies two tracking routines after each measurement; when it is set to FAST, it uses only one tracking routine; when it is set to OFF, it does not use either of the tracking routines.

COMMENTS
- *RST: FAST
- The SLOW setting provides the most accurate signal tracking but yields the smallest number of measurements in a given period of time.
- The FAST setting centers the IF in the IF bandwidth after each measurement to improve tracking.

RELATED FRONT-PANEL KEYS None

[:SENSe]:FUNCtion Subtree

This subtree controls the sensor functions.

COMMAND **[:SENSe]:FUNCtion[:OFF] . . . <sensor_function>[,<sensor_function>]**

Sets or queries the sensor functions to be sensed by the Counter.

The <sensor_function> strings are:

"[XNONe:]FREQuency [1 | 2]"

"[XNONe:]POWer [2]"

QUERY RESPONSE

- The query form of this command returns a comma-separated list of functions that are OFF.
- The string omits default nodes (XNONe) and uses short-form mnemonics. If the channel specifier(s) are set to default value(s), no channel specifier is returned in response. If the channel specifier(s) are not set to default value(s), they are returned in the response with a single space separating the first channel specifier from the function name.

  For example:

  - "FREQ" is returned for frequency on Channel 1.
  - "FREQ 2" is returned for frequency on Channel 2.

COMMENTS

- *RST: "FREQ 2"
- This command can be used to turn individual function(s) OFF without affecting other functions.
- If the optional channel specification is omitted from the <sensor_function>, a default channel selection is made. For Frequency and Power, the default is Channel 2.

RELATED FRONT-PANEL KEYS

Various

COMMAND **[:SENSe]:FUNCtion[:ON] . . . <sensor_function>[,<sensor_function>]**

Selects the sensor functions to be sensed by the Counter.

The supported <sensor_function> strings are:

"[XNONe:]FREQuency [1 | 2]"

"[XNONe:]POWer [2]"

QUERY RESPONSE

- The query form of this command returns a comma-separated list of functions that are ON.
- The string omits default nodes (XNONe) and uses short-form mnemonics. If the channel specifier(s) are set to default value(s), no channel specifier is returned in response. If the channel specifier(s) are not set to default value(s), they are returned in the response with a single space separating the first channel specifier from the function name.

  For example:

  – "FREQ" is returned for frequency on Channel 1.

  – "FREQ 2" is returned for frequency on Cbannel 2.
- The only functions that can be turned ON simultaneously are "POW 2" and "FREQ 2".

COMMENTS

- *RST: "FREQ 1"
- This command can be used to turn individual function(s) ON without affecting other functions.
- If the optional channel specification is omitted from the <sensor_function>, a default channel selection is made. For Frequency and Power, the default is Channel 2.

RELATED FRONT-PANEL KEYS

Various

COMMAND **[:SENSe]:FUNCtion:STATe? . . . <sensor_function>**

This query-only command returns a Boolean value that indicates whether the specified <sensor_function> is currently ON or OFF.

COMMENT See [:SENSe]:FUNCtion[:ON] on page 4-28 for valid <sensor_function> strings.

[:SENSe]:POWer Subtree

This subtree controls the power-measurement function.

COMMAND **[:SENSe]:POWer:AC:REFerence . . . <numeric_value><power_units>**

This command sets a reference amplitude (in dB) for display of power measurements. It is intended for use as a measurement offset, so that a measurement can be referenced to a known value.

RANGE –50 dBm to 10 dBm

UNITS dB, dBm

COMMENT *RST: 0

RELATED FRONT-PANEL KEYS **Shiff + Pwr Offset**

COMMAND **[:SENSe]:POWer:AC:REFerence:STATe . . . <Boolean>**

Determines whether amplitude is measured in absolute or relative mode. If STATe is ON, amplitude is referenced to the value set in REFerence.

COMMENT *RST: OFF

RELATED FRONT-PANEL KEYS Power **Offset On/Off**

[:SENSe]:ROSCillator Subtree

This subtree controls the Reference Oscillator.

COMMAND **[:SENSe]:ROSCillator:SOURce . . . <character_program_data>**

Sets or queries the current reference timebase to INTernal or EXTernal.

QUERY RESPONSE A sequence of ASCII-encoded bytes: INT or EXT

COMMENTS
- *RST: INT
- INTernal indicates the timebase is the internal reference. EXTernal indicates the signal at the external reference input (located on the rear panel of the Counter; **Reference** connector) is the reference timebase.
- The Counter does not switch to EXTernal unless a suitable 1, 2, 5, or 10 MHz signal is present on the "Reference" connector on the back panel.
- Execution of the command (that is, explicitly selecting internal or external timehase) sets [:SENS]:ROSC:SOUR:AUTO to OFF.
- If this is set toEXT, and no valid external signal is available at the back-panel Reference connector, the front-panel frequency display may show an error message.
- If this is set to INT, the 10 MHz signal generated by the internal reference oscillator is availahle as an output on the back-panel Reference connector.

RELATED FRONT-PANEL KEYS **Shift** + **Menu** (REF OSC > INT | EXT)

:STATus Subsystem

The :STATus subsystem commands allow you to specify or examine the status of the Operation Status Register group and the Questionable Data/Signal Register group.

:STATus:OPERation Subtree

The :STATus:OPERation subtree commands allow you to examine the status of the Counter monitored by the Operation Status Register Group, shown in Figure 4-1. The Operation Status Register Group consists of a condition register, two transition registers, an event register, and an enable register. Tbe commands in this subtree allow you to control and monitor these registers.

See the sections titled "Operation Status Register Group" and "Questionable Data Status Register Group" on pages 3-37 and 3-39 for a detailed description of the Operation Status Register Group.

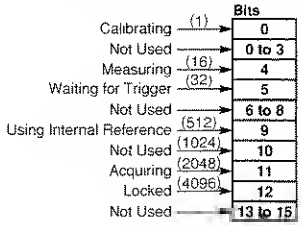


Figure 4-1. The Operation Status Register Group

COMMAND **:STATus:OPERation:CONDition?**

Queries the status of the Operation Condition Status Register.

QUERY RESPONSE
- Numeric data transferred as ASCII hytes in <NR1> format.
- Range is 0 to 32,767.
- The query response value is an integer formed hy the binary weighting of the bits. The value of unused bits is zero.

COMMENTS
- Query only.
- The Operation Condition Status Register is cleared at power-on.
- Bits are not cleared when read.

RELATED FRONT-PANEL KEYS None

COMMAND **:STATus:OPERation:ENABle <non-decimal numeric> | <NRf>**

Sets or queries the Operation Event Status Enable Register.

RANGE The range for the <non-decimal numeric> or <NRf> parameter is 0 to 32,767.

QUERY RESPONSE Numeric data transferred as ASCII bytes in <NR1> format.

COMMENTS
- The parameter and query response value, when rounded to an integer value and expressed in base 2 (binary), represent the bit values of the Operation Event Status Enable Register.
- The value of unused bits is zero when queried and is ignored when set.
- This register is used to enable a single or inclusive OR group of Operation Event Status Register events to be summarized in the Status Byte Register (bit 7).
- At power-on and :STAT:PRES, the Operation Event Status Enable Register is cleared (value is 0).
- This value is unaffected by *RST and *SAV/*RCL.

RELATED FRONT-PANEL KEYS None

COMMAND **:STATus:OPERation[:EVENt]?**

Queries the status of the Operation Event Status Register.

QUERY RESPONSE

- Numeric data transferred as ASCII bytes in <NR1> format.
- Range is 0 to 32,767.
- The query response value is an integer formed by the binary weigbting of bits. The value of unused bits is zero.

COMMENTS

- Each event bit in the Operation Event Status Register corresponds to a specific condition bit in the Operation Condition Status Register; this allows the Operation Event Status Register to detect changes in conditions.
- An event becomes TRUE when the associated condition makes the transition specified by the transition filters.
- The event bits, once set, are "sticky"—i.e., they cannot be cleared until they are read, even if they do not reflect the current status of a related condition.
- The Operation Event Status Register is cleared by *CLS, by :STAT:OPER[:EVEN]?, and at power-on.

RELATED FRONT-PANEL KEYS

None

COMMAND **:STATus:OPERation:NTRansition <non-decimal numeric> | <NRf>**

Sets or queries the negative transition filter for the Operation status reporting structure.

RANGE The range of the <non-decimal numeric> or <NRf> parameter is 0 to 32,767.

QUERY RESPONSE Numeric data transferred as ASCII bytes in <NR1> format.

COMMENTS
- The parameter and the query response value, when rounded to an integer value and expressed in base 2 (binary), represent the bit values of the negative transition filter.
- The value of unused bits is zero when queried and is ignored when set.
- A TRUE bit in the negative transition filter specifies that a negative (TRUE to FALSE) transition of the corresponding bit in the Operation Condition Status Register generates the corresponding event in the Operation Event Status Register.
- At power-on and STAT:PRES, the negative transition filter is preset such that each bit is a 0 (FALSE).
- This value is unaffected by *RST, *CLS, and *SAV/*RCL.

RELATED FRONT-PANEL KEYS None

COMMAND **:STATus:OPERation:PTRansition . . . <non-decimal numeric> | <NRf>**

Sets or queries the positive transition filter for the Operation status reporting structure.

RANGE The range of the <non-decimal numeric> or <NRf> parameters is 0 to 32,767.

QUERY RESPONSE Numeric data transferred as ASCII bytes in <NR1> format.

COMMENTS
- The parameter and the query response value, when rounded to an integer value and expressed in base 2 (binary), represent the bit values of the positive transition filter.

- The value of unused bits is zero when queried and is ignored when set.
- A TRUE bit in the positive transition filter specifies that a positive (FALSE to TRUE) transition of the corresponding bit in the Operation Condition Status Register generates the corresponding event in the Operation Event Status Register.
- At power-on and STAT:PRES, the positive transition filter is preset such that each bit is a 1 (TRUE).
- This value is unaffected by *RST, *CLS, and *SAV/*RCL.

RELATED FRONT-PANEL KEYS: None

COMMAND **:STATus:PRESet**

This event command presets the enable registers and transition filters associated with the Operation and Questionable status reporting structures. The enable registers and negative transition filters are preset such that each bit is a 0 (FALSE). The positive transition filters are preset such that each bit is a 1 (TRUE).

:STATus:QUEStionable Subtree

The :STATus:QUEStionable subtree commands allow you to examine the status of the Counter monitored by the Questionable Data/Signal Status Register Group, shown in Figure 4-3. The Questionable Status Group consists of a condition register, two transition registers, an event register, and an enable register. The commands in this subtree allow you to control and monitor these registers.

See the sections titled "Operation Status Register Group" and "Questionable Data Status Register Group" on pages 3-37 and 3-39 for a detailed description of the Questionable Data/Signal Status Register Group.

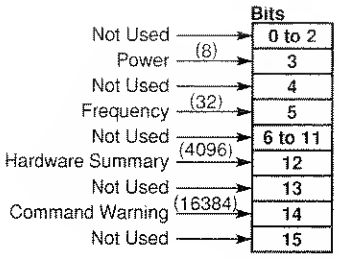


Figure 4-2. The Questionable Data/Signal Status Register Group

COMMAND **:STATus:QUEStionable:CONDition?**

Queries the status of the Questionable Data Condition Status Register.

QUERY RESPONSE

- Numeric data transferred as ASCII bytes in <NR1> format.
- Range is 0 to 32,767.
- The query response value is an integer formed by the binary weighting of the bits. The value of unused bits is zero.

COMMENTS

- The Questionable Data Condition Status Register is cleared at power-on.
- Bits are not cleared when read.

RELATED FRONT-PANEL KEYS: None

COMMAND **:STATus:QUEStionable:ENABle . . . <non-decimal_numeric> | <NRf>**

Sets or queries the Questionable Data Event Status Enable Register.

RANGE: The range of the <non-decimal numeric> or <NRf> parameter is 0 to 32,767.

QUERY RESPONSE: Numeric data transferred as ASCII bytes in <NR1> format.

COMMENTS:

- The parameter and the query response value, when rounded to an integer value and expressed in base 2 (binary), represent the bit values of the Questionable Data Event Status Enable Register.
- The value of unused bits is zero when queried and ignored when set.
- This register is used to enable a single or inclusive OR group of Questionable Data Event Status Register events to be summarized in the Status Byte Register (bit 3).
- At power-on and :STAT:PRES, the Questionable Data Event Status Enable Register is cleared (value is 0).
- This value is unaffected by *RST, *CLS, and *SAV/*RCL.

RELATED FRONT-PANEL KEYS: None

COMMAND **:STATus:QUEStionable[:EVENt]?**

Queries the status of the Questionable Data Event Status Register.

QUERY RESPONSE:

- Numeric data transferred as ASCII bytes in <NR1> format.
- Range is 0 to 32,767.
- The query response value is an integer formed by the binary weighting of bits. The value of unused bits is zero.

COMMENTS:

- Each event bit in the Questionable Data Event Status Register corresponds to a specific condition bit in the Questionable Data

Condition Status Register; this allows the Questionable Data Status Register to detect changes in conditions.

- An event becomes TRUE when the associated condition makes the transition specified by the transition filters.
- The event bits, once set, are "sticky"—i.e., they cannot be cleared until they are read, even if they do not reflect the current status of a related condition.
- The Questionable Data Event Status Register is cleared by *CLS, by :STAT:QUES[:EVEN]?, and at power-on.

RELATED FRONT-PANEL KEYS: None

:SYSTem Subsystem

This subsystem collects together the capabilities that are not related to instrument performance.

:SYSTem:COMMunicate Subtree

The :SYSTem:COMMunicate subtree collects together the configuration of the control/communication interfaces.

The :SYSTem:COMMunicate:SERial subtree controls the physical configuration of the RS-232C port. Any command to change the settings takes effect immediately upon receipt of the "program message termination." These settings are stored in non-volatile memory, and are unaffected by power-on, *SAV/*RCL, and *RST.

The :SYSTem:COMMunicate:SERial:TRANsmit subtree controls parameters associated with transmission.

The Counter always uses eight data bits, one stop bit, and no parity.

COMMAND **:SYSTem:COMMunicate:GPIB[:SELF]:ADDress . . . <numeric_value>**

Sets the HP-IB address that the Counter uses.

REPLY FORMAT <NR1>

COMMENTS
- The range for <numeric_value> is 0 through 30.
- The default setting is address 19.
- This value is unaffected by *RST.
- To change the HP-IB address used at power-on, save the instrument settings using the *SAV 0 command after changing the HP-IB address.

RELATED FRONT-PANEL KEYS **Shift + HPIB**

COMMAND **:SYSTem:COMMunicate:SERial:BAUD . . . <numeric_value>**

Sets or queries the baud rate.

<NUMERIC_VALUE> RANGE The possible BAUD rate values that can be entered for the <numeric_value> parameter are: 1200, 2400, 4800, 9600, 14400, and 19200.

QUERY RESPONSE Numeric data transferred as ASCII bytes in <NR1> format.

COMMENTS
- This value is unaffected by *RST.
- To change the Baud rate used at power-on, save the instrument settings using the *SAV 0 command after changing the Baud rate.

RELATED FRONT-PANEL KEYS **Shift + Menu** (BAUD > 19200 | 14400 | 9600 | 4800 | 2400 | 1200)

COMMAND **:SYSTem:ERRor?**

Queries the oldest error in the Error Queue and removes tbat error from the queue (first in, first out).

See Chapter 5, "Errors," for detailed error information

QUERY RESPONSE
- The response is in the following form: <error_number>,"<error_description>"
- The <error_number> is an integer in the range [-32768, 32767]. The negative error numbers are defined by the SCPI standard; positive error numbers are particular to this Counter. An error number value of zero indicates that the Error Queue is empty.
- The maximum length of the <error_description> is 255 characters.

COMMENTS

- The queue is cleared (emptied) on *CLS, power-on, or upon reading he last error from queue.
- If the Error Queue overflows, the last error in the queue is replaced with the error -350, "Queue overflow". Any time the queue overflows, the least recent errors remain in the queue and the most recent error is discarded. The maximum length of the Error Queue is 30.
- The Error Queue is unaffected by *RST and *SAV/RCL. It is cleared by *CLS.

RELATED FRONT-PANEL KEYS: None

COMMAND **:SYSTem:KEY:CATalog?**

This query returns a list of the Counter's keys that have command equivalents. The response is a definite-length block that lists the key codes and their meanings. There is no command version.

RELATED FRONT-PANEL KEYS: None

COMMAND **:SYSTem:KEY . . . <numeric_value>**

This command simulates the pressing of a front-panel key. The <numeric_value> is a key-code value.

This command puts an entry in the Key Queue (just as any front-panel key-press does). There is no query version of this command.

The keys and their corresponding key codes are listed in Table 4-3.

Table 4-3. Front Panel Keys and Key Codes

Key Name	Key Code	Key Name	Key Code
Reset/Local	7	Chan Select	0
Rate	2	Display Power	8
Avg	6	Offset On/Off (Frequency)	4
Resol	10	Offset On/Off (Power)	12
Shift	11	Up Arrow	1
Clear	14	Down Arrow	9
+/–	3	Left Arrow	5
Enter	15	Right Arrow	13

<NUMERIC_VALUE> RANGE

1 to 15

COMMENTS

- The Key Queue is unaffected by *SAV/*RCL.
- Key commands are sequential, but only in terms of processing other key commands or getting into the Key Queue. The operation performed by the key command is not guaranteed to be complete before processing of the next non-key command. Be aware of this when intermixing key commands and non-key commands.

RELATED FRONT-PANEL KEYS

None

COMMAND **:SYSTem:VERSion?**

Queries the SCPI version number with which the Counter complies.

OUERY RESPONSE

- Numeric data transferred as ASCII bytes in <NR2> format.
- The response is an <NR2> formatted numeric value which has the form YYYY.V, where YYYY represents the year, and V represents an approved version for that year.

RELATED FRONT-PANEL KEYS

None

:TRIGger Subsystem

COMMAND **:TRIGger[SEQuence | STARt]:HOLDoff . . . <numeric_value>**

When INIT:CONT is ON, this command determines the rate at which measurements are made by setting a delay between measurements. Its setting corresponds to the front-panel rate setting as follows:

<NUMERIC_VALUE> RATE

0.0 = FAST; 0.5 = MEDium; 1.0 = SLOW

COMMENTS

- *RST: 0.0
- The only settings that are accepted are the three listed above.
- The measurement rate is also affected by the Resolution setting.
- When INIT:CONT is OFF, the holdoff has no effect.
- There is no setting for this command that corresponds to the front-panel rate setting of HOLD. The HOLD mode is entered through *RST or INIT:CONT OFF. The instrument then stops making measurements until it is triggered or until it receives INIT:CONT ON.

RELATED FRONT-PANEL KEYS

Rate

Common Commands

The IEEE 488.2 Common Commands are general-purpose commands that are common to all instruments (as defined in IEEE 488.2). These commands are generally not related to measurement configuration. They are used for functions like resetting the instrument, identification, or synchronization.

*CLS
(Clear Status Command)

COMMAND ***CLS**

Clears all event registers summarized in the status byte (Standard Event Status Register, Operation Event Status Register, and Questionable Data Event Status Register) and clears the Error Queue.

COMMENTS

- The *CLS command does not clear data memories or any other settings.
- *CLS places the instrument in "Operation Complete Idle State" and "Operation Complete Query Idle State" (IEEE 488.2). This results in the disabling of any prior *OPC command.
- If *CLS immediately follows a program message terminator, the output queue and the MAV bit are cleared, since any new program message after a program-message terminator clears the output queue.
- This command clears any displayed error message from the front panel.
- If the front panel is displaying any menu when *CLS is received, the menu is abandoned (equivalent to pressing the **Clear** key).

RELATED FRONT-PANEL KEYS **Clear**

*DDT <arbitrary block> (Define Device Trigger Command) *DDT? (Define Device Trigger Query)

COMMAND ***DDT**
***DDT?**

Sets or queries the action that the device executes when it receives the IEEE 488.1 Group Execute Trigger (GET) interface message (page 4-7) or a *TRG common command.

QUERY RESPONSE

- Definite length block
- The query response is one of the following terminated with a new line and EOI:

  #14INIT
  #216INIT:*WAI;:DATA?
  #0
  #15FETC?
  #15READ?

COMMENTS

- *RST: #18INIT:1MM
- If a zero-length <arbitrary block> is specified as tbe parameter, the Counter does nothing when it receives a GET or *TRG command.

RELATED FRQNT-PANEL KEYS None

*ESE
(Standard Event Status Enable Command)
*ESE?
(Standard Event Status Enable Query)

COMMAND ***ESE <NRf> | <non-decimal numeric>**
***ESE?**

Sets or queries the Standard Event Status Enable Register, shown in Figure 4-3.

The parameter and query response value, when rounded to an integer value and expressed in base 2 (binary), represents the bit values of the Standard Event Status Enable Register. The value of unused bits is zero when queried and ignored when set.

This register is used to enable a single or inclusive OR group of Standard Event Status Register events to be summarized in the Status Byte Register (bit 5).

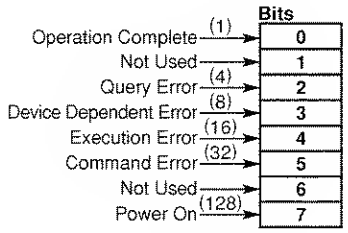


Figure 4-3. The Standard Event Status Enable Register

See the section titled "Standard Event Status Register" on page 3-30 of this guide for a detailed description of the Standard Event Status Register.

<NRF> RANGE 0 to 255

<NRF> RESOLUTION 1

QUERY RESPONSE Numeric data transferred as ASCII bytes in <NR1> format.

COMMENTS
- At power-on, the Standard Event Status Enable Register is cleared (value is 0).
- This value is unaffected by *RST and *SAV/*RCL.
- Values for *ESE may be entered as decimal, hexadecimal, octal, or binary numbers.

RELATED FRONT-PANEL KEYS None

*ESR?
(Event Status Register Query)

COMMAND ***ESR?**

Queries the Standard Event Status Register, shown in Figure 4-4.

This event register captures changes in conditions, by having each event bit correspond to a specific condition in the instrument. An event becomes TRUE when the associated condition makes the defined transition. The event bits, once set, are "sticky"—i.e., they cannot be cleared until they are read, even if they do not reflect the current status of a related condition.

This register is cleared by *CLS, by *ESR?, and at power-on. Note that the instrument's power-on sequence initially clears the register, but then records any subsequent events during the power-on sequence including setting the PON (power on) bit.

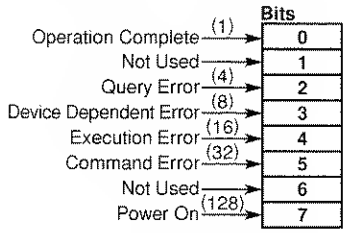


Figure 4-4. Standard Event Status Register

See the section titled "Standard Event Status Register" on page 3-30 of this guide for a detailed description of the Standard Event Status Register.

QUERY RESPONSE

- Numeric data transferred as ASCII bytes in <NR1> format.
- Range is 0 to 255.
- The query response is an integer formed by the binary-weighting of the bits. The value of any unused bit is zero.

*IDN?
(Identification Query)

COMMAND ***IDN?**

Queries the Counter identification.

QUERY RESPONSE A sequence of ASCII-encoded bytes:

HEWLETT PACKARD, <Model Number>, <Serial Number>, <Firmware ID>

COMMENTS
- This query should be the last query in a terminated program message; if it is not the last query, an error -440 is generated.
- The model number is either 53150A, 53151A, or 53152A.
- The number of digits in the serial number is not fixed.
- The format for the firmware ID is: H0-nnn, where nnn is a three-digit number. This is followed by the date and time of the firmware release.

RELATED FRONT-PANEL KEYS **Shift + Menu**

*IST?
(Instrument Status)

COMMAND ***IST?**

Queries the current state of the parallel poll response (Instrument Status).

OUERY RESPONSE <NR1>

RANGE 0-1

RELATED FRONT-PANEL KEYS None

*OPC
(Operation Complete Command)

COMMAND ***OPC**

This event command enables the OPC bit (bit 0) in the Standard Event Status Register to be set when a triggered action is complete. See the section titled "Standard Event Status Register" on page 3-30 of this guide for a detailed description of the Standard Event Status Register's Operation Complete bit.

This event command is "disabled" by *CLS, *RST, power-on, or upon the transition of the measurement cycle from measuring to idle.

This event command has no query form.

See the section titled "Using the *OPC Command to Assert SRQ" on page 3-50 for an example that uses this command.

RELATED FRONT-PANEL KEYS: None

*OPC?
(Operation Complete Query)

COMMAND ***OPC?**

This query causes the instrument to place a response in the output queue when a triggered action is complete. This allows synchronization between a controller and the instrument using the MAV bit in the Status Byte Register. (Note that this query does not actually "read" a state, as most queries do.)

See the section titled "Using the *OPC? Command" on page 3-49 for an example that uses this command.

QUERY RESPONSE: Single ASCII-encoded byte, 1.

RELATED FRONT-PANEL KEYS: None

NOTE

The *OPC? query does not in any way affect the OPC bit in the Standard Event Status Register.

*PRE
(Parallel Poll Enable Register)
PRE?
(Parallel Poll Enable Register Query)

COMMAND ***PRE<NRf>**
***PRE?**

Sets or queries the value of the parallel poll enable register.

QUERY RESPONSE <NRf>

RANGE 0-255

COMMENTS
- The parallel poll enable register is eight bits wide and has the same bit definitions as the status byte.
- The status byte and parallel poll enable registers are anded together; the result determines the value of *IST (TRUE or FALSE).

RELATED FRONT-PANEL KEYS None

*RCL
(Recall Command)

COMMAND ***RCL <NRf> | <non-decimal numeric>**

This command restores the state of the instrument from a copy stored in local non-volatile memory. Before the recall occurs, the current state of the instrument is automatically saved to register 0.

<NRF> RANGE 0 to 8

<NRF> RESOLUTION 1

RELATED FRONT-PANEL KEYS **Shift + Menu** (RECALL > 0 through 8)

*RST (Reset Command)

COMMAND ***RST**

This event command performs an instrument reset.

The reset performs the following:

- sets instrument settings to their *RST states, and
- places the instrument in "Operation Complete Idle State" and "Operation Complete Query Idle State."

COMMENTS

- See the section titled "*RST Response" on page 2-19 for a complete listing of the *RST state.
- Each command description in this chapter includes the *RST state in the "Comment" portion of the definition.

RELATED FRONT-PANEL KEYS **Reset/Local**

*SAV
(Save Command)

COMMAND ***SAV <NRf> | <non-decimal numeric>**

This command stores the current state of the instrument in register 0 of local non-volatile memory when *RCL or is executed or Recall is selected from the menu using the front-panel controls.

<NRF> RANGE 1 to 8

<NRF> RESOLUTION 1

COMMENTS

- The following states are saved:

:DISPlay[:WINDow]:BACKground:[:STATe]
:INITiate:CONTinous
[:SENSe]:AVERage[:STATe]
[:SENSe]:AVERage:COUNt
[:SENSe]:CORRection:CSET:SELect
[:SENSe]:CORRection:CSET:STATe
[:SENSe]:FILTer:FM:AUTO
[:SENSe]:FREQuency:OFFset:STATe
[:SENSe]:FREQuency:RESolution
[:SENSe]:FREQuency:TRACking
[:SENSe]:FUNCtion[:ON]
[:SENSe]:POWer:AC:REFerence
[:SENSe]:POWer:AC:REFerence:STATe
[:SENSe]:ROSCillator:SOURce
:TRIGger:[:SEQuence]:HOLDoff
*DDT

- The following front-panel settings are saved (these settings have no command equivalents):

Rel Pwr (Relative Power) value
Rel Pwr (Relative Power) state
Power Display Units
Rel Freq (Relative Frequency) value
Rel Freq (Relative Frequency) state

RELATED FRONT-PANEL KEYS **Shift + Menu** (SAVE > 0 through 8)

*SRE (Service Request Enable Command) *SRE? (Service Request Enable Query)

COMMAND ***SRE <NRf> | <non-decimal numeric>**
***SRE?**

Sets or queries the Service Request Enable Register, which is shown in Figure 4-5.

The parameter and query response value, when rounded to an integer value and expressed in base 2 (binary), represents the bit values of the Service Request Enable Register.

This register is used to enable a single or inclusive OR group of Status Byte Register events to generate an SRQ.

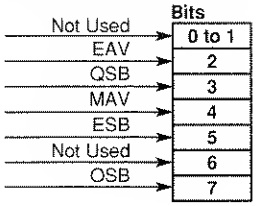


Figure 4-5. The Service Request Enable Register

See the section titled "Status Byte Register and Service Request Enable Register" starting on page 3-27 for a detailed description of the Service Request Enable Register.

<NRF> RANGE
- 0 to 255
- The value of bit 6 is ignored.

<NRF> RESOLUTION 1

QUERY RESPONSE
- Numeric data transferred as ACSII bytes in <NR1> format.
- The value of bit 6 is zero when queried.

COMMENTS
- At power-on, this value is cleared (set to 0).
- This value is unaffected by *RST, *CLS, and *SAV/*RCL.

RELATED FRONT-PANEL KEYS
None

*STB?
(Status Byte Query)

COMMAND ***STB?**

Queries the Status Byte Register, shown in Figure 4-6.

This register is cleared at power-on.

This query does not directly alter the Status Byte Register (including the MSS/RQS bit) or anything related to the generation of SRQ.

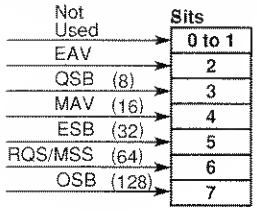


Figure 4-6. The Status Byte Register

See the section titled "Status Byte Register and Service Request Enable Register" starting on page 3-27 for a detailed description of the Status Byte Register.

QUERY RESPONSE

- Numeric data transferred as ASCII bytes in <NR1> format.
- Range is 0 to 255.
- The response value when rounded to an integer value and expressed in base 2 (binary), represents tbe bit values of the Status Byte Register.
- The value of unused bits is zero when queried.
- The Master Summary Status, not the RQS message, is reported on bit 6. Master Summary Status indicates that the Counter has at least one reason for requesting service. (The Master Summary Status is not sent in response to a serial poll; the IEEE 488.1 RQS message is sent instead.) It is the inclusive OR of the bitwise combination (excluding bit 6) of the Status Byte Register and the Service Request Enable Register.

*TRG
(Trigger Command)

COMMAND ***TRG**

This command is the device-specific analog of the IEEE 488.1 Group Execute Trigger (GET) interface message (page 4-7), and has exactly the same effect.

The *TRG command performs the action defined by the *DDT command (page 4-46).

RELATED FRONT-PANEL KEYS None

*TST?
(Self-Test Query)

COMMAND ***TST?**

This query causes an internal self-test; the response indicates whether any errors were detected.

When the self-test fails, error -330 is generated, and other messages indicating specific failures are also placed in the error/event queue.

QUERY RESPONSE

- Numeric data transferred as ACSII bytes in <NR1> format.
- A response value of zero indicates the self-test has completed with no errors detected, while a non-zero value indicates the self-test was not completed or was completed with errors detected. The test failures that can be detected and their corresponding bit numbers are shown below:

Table 4-4. Self-Test Error Values

Bit	Test Failed	Bit	Test Failed
0	Band 1 Signal Path	19	ROM
1	Band 1 Threshold	20	Unused
2	Band 2 RF Threshold	21	Front Panel Hardware
3	Band 2 IF Through Threshold	22	HP-IB
4	Band 2 IF Heterodyne Threshold	23	EEPROM Instrument Configuration Data
5	Heterodyne Path	24	EEPROM Service Data
6	Through Path	25	EEPROM Saved User Settings Data
7	VCO	26	EEPROM Power Calibration Data
8	Counter Control FPGA	27	EEPROM Write
9	Power Measurement Hardware	28	EEPROM Power Correction Data
10-15	Unused	29	ADC
16	–12 V	30	Over Temperature
17	–5 V	31	Unused
18	+12 V		

- The decimal weight of each bit is 2^n, where n is the bit number.

COMMENTS The following are tested:

CPU	EEPROM
Front-panel components	Measurement hardware
ROM	Power supply outputs

RELATED FRONT-PANEL KEYS **Shift** + **Menu** (DO SELF TEST)

*WAI
(Wait-to-Continue Command)

COMMENTS ***WAI**

This command prevents the instrument from executing any further commands or queries until all pending operations are complete. The only way to cancel this "holdoff" is by device clear, power-on, *RST, or *CLS.

See the section titled "Using the *WAI Command" on page 3-49 for an example that uses this command.

RELATED FRONT-PANEL KEYS

None

5

Errors

Introduction

This chapter explains how to read error messages from the Counter, discusses the types of errors, and provides a table of all of the Counter's error messages and their probable causes.

Reading an Error

Executing the :SYSTem:ERRor? command reads the oldest error from the error queue and erases that error from the queue. The :SYST:ERR? response has the form:

<error number>, <error string>

An example response is:

-113,"Undefined header"

All errors set a corresponding bit in the Standard Event Status Register (see the section titled "Standard Event Status Register Group" on page 3-30).

The following short program reads all errors (one at a time, oldest to newest) from the error queue. As each error is read, it is automatically erased from the error queue. When the error queue becomes empty (that is, when all errors have been read from the queue), further queries return the **+0, "No error"** response.

```
10 ASSIGN @Cntr TO 703
20 !Assign path name
30 DIM Err_string$[255]
40 !Creates array for error string
50 REPEAT
60 !Repeats until error queue is empty
70 OUTPUT @Cntr;"SYST:ERR?"
80 !Read error number and string
90 ENTER @Cntr;Err_num,Err_string$
100 !Enter error number and string
110 PRINT Err_num,Err_string$
120 !Print error number and string
130 UNTIL Err_num = 0
140 END
```

Error Queue

As errors are detected, they are placed in an error queue. The error queue is a first in, first out queue. That is, if more than one error has occurred, the first error in the queue is read out with :SYST:ERR?. Subsequent responses to :SYST:ERR? continue until the queue is empty.

If the error queue overflows, the last error in the queue is replaced with error –350, "Queue overflow". Any time the queue overflows, the least recent errors remain in the queue, and the most recent error is discarded. The length of the Counter's error queue is 10 (9 positions for the error messages, and 1 position for the "Queue overflow" error). Reading an error from the head of the queue removes that error from the queue, and opens a position at the tail of the queue for a new error, if one is subsequently detected. When all errors have been read from the queue, further error queries return **+0, "No error"**.

The error queue is cleared when any of the following events occur:

- Power-on.
- Receipt of a *CLS command.
- The last item is read from the queue.

Error Types

Error numbers are categorized by type as shown in Table 5-1. The error codes that can be generated by the HP 53151A/152A/153A Counters are listed in Table 5-2.

Table 5-1. Error Types

Error Number	Error Type
+0	No Error
–100 to –199	Command Errors
–200 to –299	Execution Errors
–300 to –350	Device-Specific Errors
–400 to –499	Query Errors

The first error described in each class (for example, -100, -200, -300, -400) is a "generic" error.

No Error

The :SYST:ERR? response **+0, "No error"** indicates that there are no errors in the Counter's error queue. The error queue is empty when every error in the queue has been read (:SYST:ERR? query) or the queue was cleared hy power-on or *CLS.

Command Error

An <error number> in the range [–100 to –199] indicates that an IEEE 488.2 syntax error was detected by the Counter's parser. The occurrence of any error in this class causes the command error bit (bit 5) in the Event Status Register to be set. This happens when one of the following events occurs:

- An IEEE 488.2 syntax error is detected by the Counter's parser. That is, a controller-to-Counter message was received that is in violation of the IEEE 488.2 Standard. Possible violations include a data element that violates the Counter listening formats or whose type is unacceptable to the Counter.
- An unrecognized header was received. Unrecognized headers include incorrect Counter-specific headers and incorrect or unimplemented IEEE 488.2 Common Commands.
- A Group Execute Trigger (GET) was entered into the input buffer inside of an IEEE 488.2 program message.

Events that generate command errors do not generate execution errors, device-specific errors, or query errors.

Execution Error

An <error number> in the range [–200 to –299] indicates that an error has been detected by the Counter's execution control block. The occurrence of any error in this class causes the execution error bit (bit 4) in the Event Status Register to be set. One of the following events has occurred:

- A <PROGRAM DATA> element following a header was evaluated by the Counter as outside of its legal input range or is otherwise inconsistent with the Counter's capabilities.
- A valid program message could not be properly executed due to some Counter condition.

Execution errors are reported by the Counter after rounding and expression evaluation operations have been taken place. Rounding a numeric data element, for example, is not reported as an execution error. Events that generate execution errors do not generate command errors, device-specific errors, or query errors.

Device- or Counter-Specific Error

An <error number> in the range [–300 to –399] or [+1 to +32767] indicates that the Counter has detected an error that is not a command error, a query error, or an execution error; some Counter operations did not properly complete, possibly due to an abnormal hardware or firmware condition. These codes are also used for self-test response errors. The occurrence of any error in this class causes the device-specific error bit (bit 3) in tbe Event Status Register to be set.

Query Error

An <error number> in the range [–400 to –499] indicates tbat the output queue control of the Counter has detected a problem with the message exchange protocol. The occurrence of any error in this class causes the query-error bit (bit 2) in the Event Status Register to be set. This means that one of the following conditions exists:

- An attempt is being made to read data from the output queue when no output is either present or pending.
- Data in the output queue was lost.

Error List

Table 5-2 lists and describes the error messages generated by the HP 53150A/151A/152A Counters.

Table 5-2. Errors

Number	Error String	Cause
+0	No error	The error queue is empty. Every error in the queue has been read (:SYSTem:ERRor? query) or the queue was cleared by power-on or *CLS.
-100	Command error	This is the generic syntax error used if the Counter cannot detect more specific errors.
-101	Invalid character	A syntactic element contains a character that is invalid for that type. For example, a header containing an ampersand, :INP:COUP& AC.
-102	Syntax error	An unrecognized command or data type was encountered.
-103	Invalid separator	The parser was expecting a separator and encountered an illegal character.
-104	Data type error	The parser recognized a data element different than one allowed. For example, numeric or string data was expected, but block data was received.
-105	GET not allowed	A Group Execute Trigger was received within a program message.
-108	Parameter not allowed	More parameters were received than expected for the header.
-109	Missing parameter	Fewer parameters were received than required for the header.
-110	Command header error	An unspecified error was detected in the header.
-111	Header separator error	A character that is not a legal header separator was encountered while parsing the header.
-112	Program mnemonic too long	The header or character data element contains more than twelve characters.
-113	Undefined header	The header is syntactically correct, but it is undefined for the Counter. For example, *XYZ is not defined for the Counter.
-114	Header suffix out of range	The value of a numeric suffix attached to a program mnemonic makes the header invalid.
-120	Numeric data error	This error, as well as errors -121 through -129, are generated when parsing a data element which appears to be numeric, including the non-decimal numeric types. This particular error message is used when the Counter cannot detect a more specific error.
-121	Invalid character in number	An invalid character for the data type being parsed was encountered. For example, a "9" in octal data.
-123	Exponent too large	Numeric overflow.
-124	Too many digits	The mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros.
-128	Numeric data not allowed	A legal numeric data element was received, but the Counter does not accept one in this position for the header.
-130	Suffix error	This error can be generated when parsing a suffix. This particular error message is used if the Counter cannot detect a more specific error (errors -131 through -139).

Table 5-2. Errors (Continued)

Number	Error String	Cause
-131	Invalid suffix	The suffix does not follow the syntax described in IEEE 488.2 or the suffix is inappropriate for the Counter.
-134	Suffix too long	The suffix contained more than 12 characters.
-138	Suffix not allowed	A suffix was encountered after a numeric element that does not allow suffixes.
-140	Character data error	This error can be generated when parsing a character data element. This particular error message is used if the Counter cannot detect a more specific error (errors -141 through -149).
-141	Invalid character data	The character data element contains an invalid character.
-144	Character data too long	The character data element contains more than twelve characters.
-148	Character data not allowed	A legal character data element was encountered where prohibited by the Counter.
-150	String data error	This error can be generated when parsing a string data element. This particular error message is used if the Counter cannot detect a more specific error.
-151	Invalid string data	A string data element was expected but was invalid for some reason. For example, an END message was received before the terminal quote character.
-158	String data not allowed	A string data element was encountered but was not allowed by the Counter at this point in parsing.
-160	Block data error	This error can be generated when parsing a block data element. This particular error message is used if the Counter cannot detect a more specific error (errors -161 through -169).
-161	Invalid block data	A block data element was expected, but it was not allowed by the Counter at this point in parsing.
-168	Block data not allowed	A legal block data element was encountered but was not allowed by the Counter at this point in parsing.
-170	Expression error	This error can be generated when parsing an expression data element. It is used if the Counter cannot detect a more specific error.
-171	Invalid expression	The expression data element was invalid (see IEEE 488.2). For example, unmatched parentheses or an illegal character.
-178	Expression data not allowed	Expression data was encountered but was not allowed by the Counter at this point in parsing.
-181	Invalid outside macro definition	Indicates that a macro parameter placeholder ($<number>) was encountered outside of a macro definition.
-200	Execution error	This is the generic syntax error if the Counter cannot detect more specific errors. This code indicates only that an Execution Error has occurred.
-210	Trigger error	Used if the Counter cannot detect a more specific error from the :INIT,:TRIG, or :ABOR subsystems.

Table 5-2. Errors (Continued)

Number	Error String	Cause
-211	Trigger ignored	Indicates that a GET or *TRG was received and recognized by the Counter but was ignored.
-213	Init ignored	Indicates that a request for a measurement initiation was ignored as another measurement was in progress.
-220	Parameter error	Indicates that a program data element related error occurred. This error is used when the Counter cannot detect more specific errors.
-221	Settings conflict	Indicates that a legal program data element was parsed but could not be executed due to the current Counter state.
-222	Data out of range	Indicates that a legal program data element was parsed but could not be executed because the interpreted value is outside the legal range defined by the Counter. Typically, the value is clipped to legal limit.
-223	Too much data	Indicates that a legal program data element of block, expression, or string type was received that contained more data than the Counter could handle due to memory or related Counter-specific requirements.
-224	Illegal parameter value	Used where exact value, from a list of possible values, was expected.
-230	Data corrupt or stale	No valid data available. New measurement started but not completed.
-240	Hardware error	Indicates that a legal program command or query could not be executed because of a hardware problem in the Counter.
-241	Hardware missing	Indicates that a legal program command or query could not be executed because of missing Counter hardware.
-300	Device-specific error	This is the generic device-dependent error.
-310	System error	Indicates that a system error occurred.
-321	Out of memory	Indicates that the Counter has detected that insufficient memory is available.
-330	Self-test failed	Indicates at least one failure occurred when *TST? was executed.
	Queue overflow	Indicates that there is no room in the error queue and an error occurred but was not recorded.
	Query error	This is the generic query error.
	Query INTERRUPTED	Indicates that a condition causing an INTERRUPTED Query error occurred. For example, a query followed by DAB or GET before a response was completely sent.
-350	Queue Overflow	Indicates that there is no room in the error queue, and that an error occurred but was not recorded.
-400	Query error	This is the generic query error.
-410	Query INTERRUPTED	Indicates that a condition causing an INTERRUPTED Query error occurred. For example, a query followed by a DAB or GET before a response was completely sent.

Table 5-2. Errors (Continued)

Number	Error String	Cause
-420	Query UNTERMINATED	Indicates that a condition causing an UNTERMINATED Query error occurred. For example, the Counter was addressed to talk and an incomplete program message was received.
-430	Query DEADLQCKED	Indicates that a condition causing a DEADLQCKED Query error occurred. For example, both input buffer and output buffer are full and the Counter cannot continue.
-440	Query UNTERMINATED after indefinite response	Indicates that a query was received in the same program message after a query requesting an indefinite response (for example, *IDN? or *OPT?) was executed.

Index

D

I

K

L

M

Technical Support:
If you need technical assistance with a Hewlett-Packard test and measurement product or application, please contact the Hewlett-Packard office or distributor in your country.

Asia Pacific:

Hong Kong
(852) 2599 7889

India
(91-11) 647.2311

Japan
Hewlett-Packard Japan Ltd.
Measurement Assistance Center
9-1, Takakura-Cho, Hachioji-Shi,
Tokyo 192, Japan
Tel: (81-426) 56-7832
Fax: (81-426) 56-7840

Korea
(82-2) 769 0800

Malaysia
(60-3) 291 0213

Philippines
(63-2) 894 1451

PRC
(86-10) 505-0149

Singapore
(1800) 292 8100

Taiwan
(886-2) 492 9666

Thailand
(66-2) 661 3900

For Countries in Asia Pacific not listed, contact:

Hewlett-Packard Asia Pacific Ltd.
17-21/F Shell Tower, Times Square,
1 Matheson Street, Causeway Bay,
Hong Kong
Tele: (852) 2599 7777
Fax: (852) 2506 9285

Australia/New Zealand:
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Australia
1 800 629 485

Canada:
Hewlett-Packard Canada Ltd.
5150 Spectrum Way
Mississauga, Ontario
L4W 5G1
(905) 206 4725

Latin America:
Hewlett-Packard
Latin American Region Headquarters
5200 Blue Lagoon Drive
9th Floor
Miami, Florida 33126
U.S.A.
(305) 267 4245/4220

United States:
Hewlett-Packard Company
Test and Measurement Organization
5301 Stevens Creek Boulevard
Building 51L-SC
Santa Clara, California 95052-8059
1 800 452 4844

Europe, Africa, and Middle East:

Austria
(0) 1 25000-0

Baltic Countries
(358) 08872 2100

Belgium and Luxembourg
(02) 778 34 17

Czech Republic
(0042) 2-4743111

Denmark
45 99 10 00

Finland
(90) 88 721

France
(1) 69.29.41.14

Germany
(0180) 532 62-33

Greece
(1) 726 4045

Hungary
(1) 252 4705

Ireland
(01) 284 4633

Israe:
(03) 5380 333

Italy
02 - 92 122 241

Netherlands
(020) 547 6669

Norway
(22) 73 57 50

Poland
(22) 508 7700

Portugal
(11) 482 85 00

Russia:
Tele: (7/095) 928-6885
Fax: (7/095) 916-9844

South Africa
(011) 806 1000

Spain
34 1 631-1323

Sweden
(08) 444 22 77

Switzerland
(01) 735 7111

Turkey
(212) 224 59 25

United Kingdom
(01344) 366 666

For countries in Europe/Middle East and Africa not listed, contact:

Hewlett-Packard
International Sales Europe
Geneva, Switzerland
Tel: +41-22-780-4111
Fax: +41-22-780-4770